EFFICIENT COMPRESSION OF SYNTHETIC VIDEO

Ahmad Abdel Jabbar Ahmad Mazhar

Faculty of Technology, De Montfort University

A thesis submitted in partial fulfillment of the requirements of

De Montfort University

for the degree of Doctor of Philosophy

February 2013

United Kingdom

Abstract

Streaming of on-line gaming video is a challenging problem because of the enormous amounts of video data that need to be sent during game playing, especially within the limitations of uplink capabilities. The encoding complexity is also a challenge because of the time delay while on-line gamers are communicating.

The main goal of this research study is to propose an enhanced on-line game video streaming system. First, the most common video coding techniques have been evaluated. The evaluation study considers objective and subjective metrics. Three widespread video coding techniques are selected and evaluated in the study; H.264, MPEG-4 Visual and VP-8. Diverse types of video sequences were used with different frame rates and resolutions. The effects of changing frame rate and resolution on compression efficiency and viewers' satisfaction are also presented. Results showed that the compression process and perceptual satisfaction are severely affected by the nature of the compressed sequence. As a result, H.264 showed higher compression efficiency for synthetic sequences and outperformed other codecs in the subjective evaluation tests.

Second, a fast inter prediction technique to speed up the encoding process of H.264 has been devised. The on-line game streaming service is a real time application, thus, compression complexity significantly affects the whole process of on-line streaming. H.264 has been recommended for synthetic video coding by our results gained in codecs comparative studies. However, it still suffers from high encoding complexity; thus a low complexity coding algorithm is presented as fast inter coding model with reference management technique. The proposed algorithm was compared to a state of the art method, the results showing better achievement in time and bit rate reduction with negligible loss of fidelity.

Third, recommendations on tradeoff between frame rates and resolution within given uplink capabilities are provided for H.264 video coding. The recommended tradeoffs are offered

as a result of extensive experiments using Double Stimulus Impairment Scale (DSIS) subjective evaluation metric. Experiments showed that viewers' satisfaction is profoundly affected by varying frame rates and resolutions. In addition, increasing frame rate or frame resolution does not always guarantee improved increments of perceptual quality. As a result, tradeoffs are recommended to compromise between frame rate and resolution within a given bit rate to guarantee the highest user satisfaction.

For system completeness and to facilitate the implementation of the proposed techniques, an efficient game video streaming management system is proposed.

Compared to existing on-line live video service systems for games, the proposed system provides improved coding efficiency, complexity reduction and better user satisfaction.

Acknowledgments

I would like to thank my supervisors. I first mention Prof. Dr. Raouf Hamzaoui due to his boundless support throughout all stages of this work. His criticism and advice provided important keys to completing this thesis. I also want to express my thankfulness to Prof Marwan Al-Akaidi for his enriching comments on the research. My sincere thanks to Dr. Ayman Abdalla as well for being my local supervisor. I thank him for his consistent support and encouraging comments during the research stages.

I also thank Dr. Shakeel Ahmad for the useful discussion on the research and for his advice on dealing with problems. I thank my colleagues Dr. Muneeb Dawood and Mr. Shadan Khattak for their cooperation and help. I want to thank my beloved De Montfort University and all its familiar staff for the interesting time I spent with them.

A special thanks to my colleagues and friends for their help and support: Dr. Osama Abdulhay for the statistical analysis and Dr. Mohammad Daoud and Mr. Hani al-Mimi for their help in the subjective evaluation. I also thank all staff and students of the Department of Multimedia Systems at Al-Zaytoonah University.

Dedication

I dedicate this work to the soul of my father who supported me during all stages of my life. I dedicate this work to my mother for encouraging and giving me the power to tackle any task during the PhD. I also dedicate the work to my brothers, sisters and fiancé for their patience and prayers. I would like also to dedicate this thesis to my uncles Dr. Abdelfatah Tamimi, Dr. Mahmoud, Eng. Jebreen, Mr. Mohammad, Dr. Abdulmoneam and Dr. Kamel for their subvention and encouragement. A very special thank to my brother-in-law Dr. Qasem Abdallah for his immeasurable support and encouragement from the first day I decided to begin my PhD.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

3G	Third Generation
ANOVA	Analysis of Variance
AVC	Advanced Video Coding
BR	Bit Rate
CIF	Common Intermediate Format
CODEC	Coder – Decoder
CT	Codec Type
dB	Decibel
DSCQS	Double Stimulus Continuous Quality Scale
DSIS	Double Stimulus Impairment Scale
FPS	Frame Per Second
HD	High Definition
HDTV	High Definition Television
HEVC	High Efficiency Video Coding
HTHM	High Texture High Motion
HTLM	High Texture Low Motion
ISO	International Organization for Standardization
JVT	Joint Video Team
LTHM	Low Texture High Motion
LTLM	Low Texture Low Motion

MC	Motion Compensation
ME	Motion Estimation
MOS	Mean Opinion Score
MPEG	Motion Picture Expert Group
MPQM	Motion Pictures Quality Metric
MSE	Mean Squared Error
NQM	Noise Quality Measure
PSNR	Peak Signal to Noise Ratio
QCIF	Quarter Common Intermediate Format
QP	Quantization Parameter
RDO	Rate-Distortion Optimization
SS	Single Stimulus
SSCQE	Single Stimulus Continuous Quality Evaluation
VCEG	Video Coding Expert Group
VQM	Video Quality Metric
VT	Video Type

1 INTRODUCTION

1.1 MOTIVATION

On-line gaming is currently one of the most popular applications. In on-line game video streaming players share the video of their game with other users. There are several online video platforms that allow users to stream the video of their game [1, 2]. An essential requirement is to capture and compress the video of the game. At the receiver side, the compressed video needs to be decompressed and displayed.

As most studies on video compression focus on natural videos, synthetic video compression and evaluation lacks research. The study of user's satisfaction with synthetic video compression is also needed.

Synthetic video may include still texture maps as well as mesh geometry for object animation and facial animation parameters. These features make synthetic videos different from natural video and give motivations for studying these video types.

One major motivation to study synthetic video compression is the varying significance of video contents in games. Unlike watching TV, gaming requires users to interact with scenes. Therefore, some contents of a synthetic video may be considered important to a user, where a similar content in natural videos appears insignificant. In this case, quality and speed are more important than other aspects of the game video. Users would prefer low frame rates with higher resolution and may prefer lower compression efficiency with high compression speed rather than high efficiency with compression delay.

Compressing and transmitting video while it is being played involves a live streaming technique and needs several steps to make it successful. The live streaming mechanism requires a video codec to compress these game videos. In addition, it needs to consider the available bandwidth capacity within the typical uplink capability offered at the present time.

The problem of compressing videos before transferring them obligates one to seek a video codec that provides high compression efficiency. Since the codec performance affects the quality of decoded videos, a comparative study of the most popular codecs is needed.

There are many video codecs available in the multimedia field, and many comparative studies have been achieved which evaluate the efficiency of video codecs for natural video sequences. However, there is a lack of research on codecs' capabilities for game videos.

The computational complexity of codecs is also an important issue. Reduction of computational complexity while preserving almost the same compression efficiency, fidelity and bit rate is a challenge in video compression. Thus, there is a need for complexity reduction techniques to speed up the encoder with minimum loss of quality and increase of bit rate.

The problem of finding a suitable tradeoff between resolution and frame rate for a given bit rate is also an essential area of study as it affects the live game video streaming. Therefore, a study of users' evaluations should provide recommendations concerning the tradeoffs between frame rates and resolutions to maximize exploiting of the available bandwidth.

The video playback delay depends on many factors such as capturing time, compression complexity, streaming capacity and decoding time. The architecture of the network and hardware used are also key factors that affect the delay.

This research targets game video streaming scenarios where there is no required response from the viewer. Therefore a delay of up to 5 seconds is generally acceptable.

1.2 PROBLEM FORMULATION

On-line video streaming applications must deal with bandwidth limitations, compression complexity and user satisfaction. Also, choosing a suitable screen resolution and providing game video smoothness are important factors in user satisfaction. Therefore, there is a need to develop a complete system to improve bandwidth exploiting of on-line streaming to increase enjoyment.

Few systems are being used for such on-line gaming services. Xfire is one of the famous systems and widely used at this time. Figure 1-1 shows the main screen in Xfire client. Users can choose friends, servers or stored files from this client interface. Figure 1- shows the games supported by Xfire. Gamers can choose the game and server, who he or she wants to play and share in this manner with other gamers.

Figure 1-1 Xfire Client's Screen

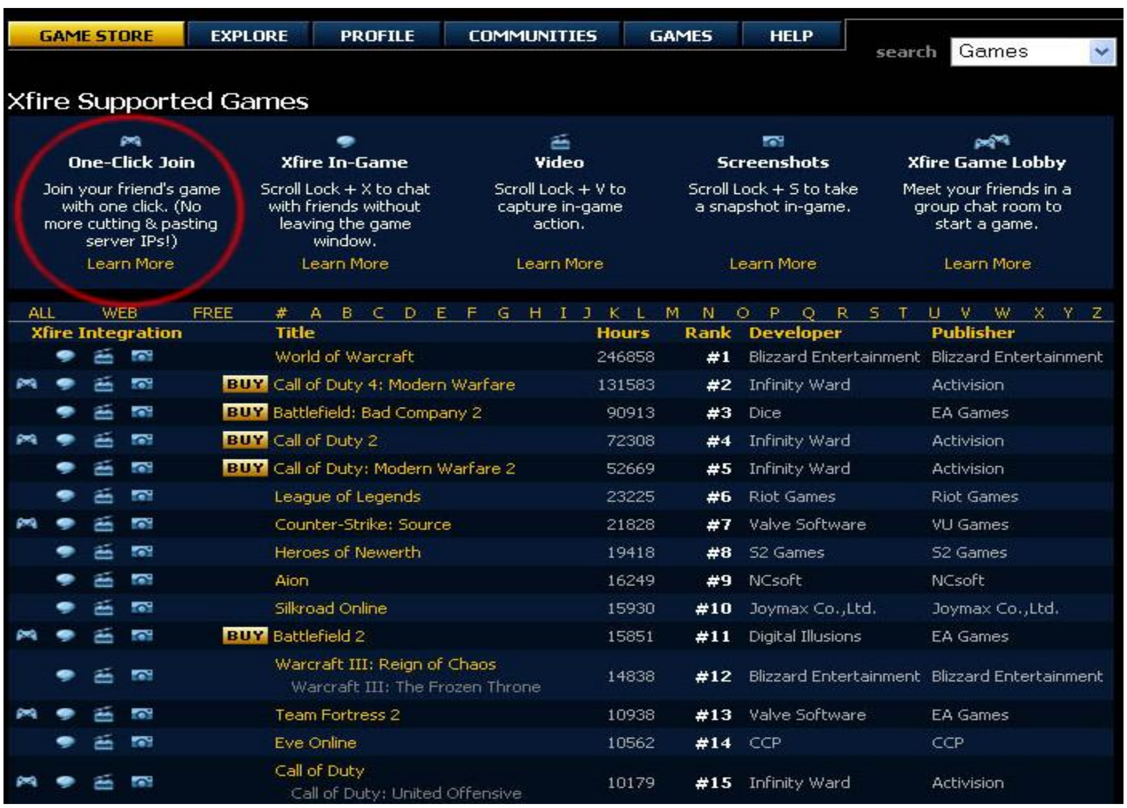

Figure 1-2 Xfire Game Store Screen

1.3 EXISTING SOLUTIONS and LIMITATIONS

According to the literature survey prepared for this research, there has been no published work on compression of game videos. The H.264/MPEG4 Advanced Video Coding standard [3] outperforms prior video compression standards by up to 50% in video compression efficiency [4]. The improvement of compression efficiency and error resilience led to increase the codec flexibility of coding and communication, which provides new efficient video services such as video telephony, wireless media streaming and on-line gaming [5]. However, adopting multiple reference frames, quarter pixel accuracy, variable block sizes and more prediction techniques has come at the expense of higher computational complexity.

Many low complexity approaches are proposed to reduce the higher complexity of the exhaustive mode decision process of H.264. The main aim of designing a new fast mode selection is a reduction of computational complexity while achieving the same level of compression. Also, the fast video compression techniques that have been proposed by many researchers have not considered the compression of synthetic videos and computer graphics.

The literature survey of this thesis shows that some previous work studied the effect of frame rate and frame resolution on perceptual satisfaction. However, the study of the tradeoff between frame rate and resolution for synthetic video types is a relatively new research area that requires further investigation.

1.4 AIMS AND OBJECTIVES

The first aim of the present research is to provide recommendations of video codecs' suitability for video game compression. The study should be able to present assessment of video quality under different spatial and temporal resolutions. Many codecs need to be evaluated under different frame rates, frame resolutions and bit rates. The evaluation should use subjective and objective assessment measurements.

The second aim of the research is to develop a novel algorithm to manage the complexity of H.264 encoder effectively. The proposed algorithm will speed up the encoding time by reducing the number of modes and reference frames required to be tested. The proposed algorithm should meet the following aims:

- It can be used in both B and P-frames.
- It should be able to reduce the computational time with minimum loss in PSNR.
- It can reduce computational time by combining two schemes, reducing number of modes needing to be checked and reducing number of reference frames needing to be estimated.
- It can be adapted with any intra frame technique.

The third aim of this research is to study the tradeoff between frame rate and resolution. Increasing frame rate and frame sizes expected to increase the overall video quality. However, such increase will be limited when the bandwidth is restricted to operate within certain levels.

The highest providers' priority is to meet user satisfaction while considering the limitation of resources. Thus, this study should present recommendations of the tradeoff between frame rates and resolution to provide highest user satisfaction within typical uplink bandwidths.

This research also presents a proposed online model system. The proposed system will exploit the previous contributions to provide a comprehensive on-line game video streaming service.

1.5 CONTRIBUTIONS OF THE THESIS

The main contributions of this thesis can be summarized as follows:

- The development of low complexity inter mode selection algorithm for an H.264 video encoder with reference frames management. A novel early prediction and termination scheme is applied to reduce the number of modes needed to be checked.

This algorithm significantly outperforms existing algorithms. This is because it is not based on any arbitrary thresholds, the algorithm is adaptive to B and P frames and because the algorithm is able to be combined with any intra frame scheme.

- A novel trade off study between frame rate and resolution. The study focuses on exploiting the typical uplink bandwidths available to meet user's satisfaction. The study presents the relationship between frame rate, frame quality and video resolution. The quality assessment has been done using one of the most popular subjective measurements, namely DSIS.
- A wide comparative study of video codecs' capabilities for game videos compression. The most popular codecs (H.264, MPEG-4 Visual and VP8) have been considered in this comparison. The novelty of this study is the idea of comparing video codecs efficiency for synthetic video sequences. The codecs' suitability has been evaluated using subjective and objective metrics.
- A survey of most of the published research on complexity reduction techniques for H.264. The survey has led to categorizing the proposed techniques in categories according to their main idea.
- A proposed model of an on-line game video streaming system. The model supports game video streaming over the Internet while taking bandwidth limitations, user satisfaction and live necessity into consideration.

1.6 OUTLINE OF THE THESIS

This dissertation is organized as follows:

Chapter 2 – Provides some essential background information on video compression techniques, video codecs and video compression for on-line gaming. The fundamental terms of video compression and main functions of typical video codecs are introduced. A brief introduction to video codecs, video quality metrics and synthetic videos are also presented.

Chapter 3 – This chapter provides a survey of previous low complexity techniques. The previous techniques are categorized and discussed. A survey of quality assessment and video gaming compression is also presented.

Chapter 4 – Presents a wide comparative study of most popular video codecs. Performance evaluations of H.264, MPEG-4 and VP-8 are evaluated with respect to different bit rates, frame rates and frame resolution. Both subjective and objective quality assessment measurements are used in this evaluation.

Chapter 5 – Proposes a fast inter mode decision with reference frame management techniques, which combines early termination of mode selection and managing the reference frames checking. The proposed algorithm achieves better results than a state of the art complexity reduction technique proposed in [6] for H.264 video codec.

Chapter 6 – This chapter demonstrates an extensive study of the tradeoff between frame rate and resolution for H.264 compressed game video. The study is conducted for several video bitrates (typical uplink bandwidths) and video types (high motion, low motion, high texture and low texture). The study is based on the DSIS subjective assessment measure which exploits users' feedback. Users' evaluations are analyzed and statistical studies are presented.

Chapter 7 – This chapter concludes the thesis and summarizes the original contributions made. The chapter also suggests certain future work in these research areas.

2 BACKGROUND

2.1 VIDEO COMPRESSION

2.1.1 Introduction

Compression is the process of compacting data into a smaller number of bits [7]. Video compression refers to the mechanism that reduces the size of data that are required to represent the video sequence or stream.

A video stream is composed of successive images that are being displayed in temporal order, each image called frame. The size of this sequence depends mainly on the nature of these frames and their number. Streams that contain frames with high numbers of pixel will require larger quantities of data to be represented [8]. Additionally, increasing number of images that are displayed per unit of time directly increases the size of the overall video stream [7, 9].

The word CODEC refers to both coder and decoder parts which are working together to compress and decompress the sequence respectively. The encoder is called the forward path while the decoder is normally called the backward path. The Encoder phase contains a reconstruction path which is responsible for decoding the block. This provides the decoded block to be referenced for the future blocks [7, 10, 11].

Limitation of resources always appears as a challenge in computer development. Video compression is a very important technique to solve the limitations of storage devices and network transmission equipment [8].

Efforts are always under study to make the compression better. Even if the resources are available to send or store the current sequences efficiently, it is better to send or store higher resolution streams by the same resources using more efficient compression techniques [9, 12].

The type of an image is a very important factor and affects the video resolution and size. Images are different in size, "size" referring in the first instance to the number of pixels that are being used to represent an image. Number of pixels is the first most important factor on image size. These numbers normally give the x and y dimensions of the image. The greater number of pixels, the higher the image resolution or the larger the image dimensions [7, 13].

The second important factor involved with image size is nature of the pixel. There are three major types of images depending on coloring; Binary, Grayscale and Colored images. For binary images, each pixel is represented by one bit zero or one. Value one represents the white color while zero represents black color. Representing each pixel by one bit, this gives these images the minimum size when compared to the others [8, 9].

The Gray-scale images require one byte to represent each pixel. One byte uses 256 bit of memory from 0 to 255. In this type of images the value 0 represents the black color and value 255 represents white. These images need more size than the binary ones [7, 12, 13].

The most popular type of images is the colored one. Three bytes are required to represent each pixel. This type of image is the most used in digital applications [7, 9].

2.1.2 Color Spaces

Two popular color spaces are being used at the present time. The first one is known as the RGB system. These letters refer to red, green and blue, which are the basic colors taken from the light spectrum. Each color component is represented by one byte. Any other color is expressed by using these basic colors [14].

The second color space is known as YCbCr. The main idea of this system is that the human eye is less sensitive to color than to brightness. Component Y refers to luminance. Cb and Cr represent the chroma components. Component Cb represents blue and Cr represents red.

Images in RGB system may be converted to YUV. It is also possible to convert YUV images to RGB. This conversion is sometimes used to convert the captured RGB images to

YUV to reduce the size needed for storage, then it is converted back to the original format when needed [10].

2.1.3 Compression Ratio

The compression ratio refers to the amount of data that is reduced. It is simply the ratio of the number of bits in the original data and the number of bits in the compressed data(2-2).

$$Compression\ Ratio = \frac{Compressed\ Data\ Rate}{Uncompressed\ Data\ Rate} \quad (2-2)$$

Two types of data compression are used, lossless and lossy compression. The techniques of lossless compression can reconstruct the original file from the compressed one. This type is used when it is very important for the decompressed file to be identical to the original source file. The most popular applications of this compression model are ZIP files [7]. It is highly used and recommended for text files and data records.

Lossy compression does not guarantee that the original and decompressed files will be identical. This type is highly used in multimedia files.

2.1.4 Bit Rate

The bit rate is the number of bits that are processed or delivered in a unit of time. Bit rates are expressed is bits per second (bps). For a high data rate application, it may be represented in kilo, mega, giga or tera (bits per second).

2.1.5 Quantization

In video compression, quantization refers to a procedure of conversion to binary with side effect of ability to reduce the amount of data that have high frequency. This procedure is applied by converting a set of continuous values (real numbers) to a discrete format (integer number) [11, 15].

One main advantage of quantization is to reduce the size of data needed to represent a part of the information. The problem is with the loss of detail. The process causes loss of some data, is thus “lossy” and is not reversible [14].

In H.264, a quantization parameter called QP is used to determine the quantization of the transform coefficients. See (2-3). The parameter can take 52 values (0 to 51). An increase of 1 in the quantization parameter means increasing the quantization step size by approximately 12%. See [7, 11].

$$FQ = round\ \frac{X}{QStep} \quad (2-3)$$

where

FQ and QStep are quantized coefficient and quantizer step size respectively.

X is the amount of data used to represent information (coefficients in video compression). This formula converts the coefficient values into quantized coefficient values.

An increase of 6 in QP means an increase the double of quantization step. The relation can be noticed as a change of step size by approximately 12% which means a reduction of bit rate by approximately 12% [7].

QP	0	1	2	3	4	5	6	7	8	9	10	11	12	…
QStep	0.625	0.6875	0.8125	0.875	1	1.125	1.25	1.375	1.625	1.75	2	2.25	2.5	…
QP	…	18	…	24	…	30	…	36	…	42	…	48	…	51
QStep		5		10		20		40		80		160		224

Figure 2-1 QP in H.264 [1]

2.1.6 Video Quality Metrics

Digital videos need to be processed in stages before they are delivered to the end-user. Most of these stages affect the video by degrading the quality, thus, video quality needs to be evaluated to determine the extent of this degradation. This evaluation is conducted using so-called video quality metrics [16].

Video quality metrics are methods used to evaluate the capability and quality of a video system. Two main types of video quality metrics are publicly used; Subjective and Objective quality metrics.

2.1.6.1 Subjective Quality Metrics

Subjective quality metrics depend mainly on human viewers to evaluate the quality. The subjective evaluation is more complicated than objective evaluation as it depends on many factors that can severely affect the judgments' credibility [16].

This type depends on measuring the compressed video sequence compared to the original one based on the human eye. This requires a certain type of users to reach the correct decision. The subjective evaluation also needs to prepare a suitable environment in which to be run, such as computers, number of trained users, suitable lighting and seating, and other such variable [17].

Subjective metrics normally use a graded scale of bad-excellent to represent the degree of quality; this is sometimes scaled at 1-5 quality values. There are several famous subjective quality metrics such as:

- DSIS (Double Stimulus Impairment Scale)
- DSCQS (Double Stimulus Continuous Quality Scale)
- SS (Single Stimulus)
- SSCQE (Single Stimulus Continuous Quality Evaluation)

All these quality metrics lead to the same target of evaluating the quality between compressed and original video sequences with certain different steps [17]. DSIS is the most popular subjective quality metric [18, 19]. Thus, it will be used in this research.

2.1.6.2 Objective Quality Metrics

Objective quality metrics is also called automatic metrics. This type is considered easier to apply than subjective metrics. The objective metrics are widely used for evaluation of video quality in research [20].

This type is conducted using some calculations that are performed by computers. The most popular objective metrics are:

- PSNR (Peak Signal to Noise Ratio)
- VQM (Video Quality Metric)
- MPQM (Motion Pictures Quality Metric)
- NQM (Noise Quality Measure)

The quality measure is one of the most important factors that should be taken into consideration in video developing. The quality measure is an indicator of the image quality. To compare two images, it is essential to calculate measures of the two [7].

The most popular measure is the Peak Signal to Noise Ratio (PSNR). It is a ratio for the fidelity between two images that normally involves the original and a decompressed version [11, 21]. Better video quality leads to gaining better PSNR. PSNR is the most common objective quality metric. Thus, it will be used in this research.

The measure uses a logarithmic scale and depends mainly on the Mean Squared Error (MSE). It gives the result as Decibel unit (dB). The higher the PSNR, the higher the quality of the decompressed image. See (2-4), where n represents the number of bits per image sample [7, 11].

$$PSNR_{dB} = 10 \log_{10} \frac{(2^{\mathrm{n}} - 1)^2}{\mathrm{MSE}} \quad (2-4)$$

2.2 VIDEO CODECS

The growth of communication networks and internet video streaming led to increasing compression services' quality. The high priority of providers is to meet users' needs by providing enjoyable quality service. High quality service offers the most powerful and clearest video quality. The increase of service quality to higher levels requires higher bit rates and consumes more bandwidth, thus, the need of video compression is required. In addition, the increment of bit rate requirements increases day by day as the applications are being developed, such as HDTV and high quality on-line gaming [8, 14].

In the field of video compression, many organizations have started developing their own codecs and compression techniques. ITU, JPEG, MPEG, ISO and IEC have been considered the most powerful standards for the last few years.

This study will investigate and compare the performance of the most popular video codecs, which provide high compression efficiency. The H.264, MPEG-4 and VP-8 video codecs are therefore introduced in this section.

2.2.1 H.264/AVC

2.2.1.1 Introduction

H.264 is the latest ITU-T and ISO/IEC video coding standard. It is used in many applications, including Blue-Ray Disc technology, IPTV, and Digital Video Broadcast. While H.264 allows better rate-distortion performance (trade-off between compression ratio and reconstruction quality) than previous standards, it has a much higher time complexity.

The H.264 is also called MPEG-4 Part 10. The goals of this standard are to enhance the compression efficiency and network suitability of video representation for both interactive and non-interactive applications [7].

H.264/AVC provides gains in compression efficiency of up to 50% over a wide range of bit rates and video resolutions compared to previous standards. The decoder is also observed to be about four times more efficient than MPEG-2 and is two times of MPEG-4 Visual Simple Profile [8, 11].

2.2.1.2 H.264 Development History

The Moving Picture Expert Group (MPEG) is the responsible for having created and updated MPEG-4. MPEG-4 is a set of standards which were created then developed to be accepted as international standards for the International Standards Organization (ISO) [21].

MPEG devised several successive standards for video and audio coding. The most widely used were MPEG-1 and MPEG-2 standards. They were highly recommended for video use over networks, for video communications and for digital video storing. MPEG-7 and MPEG-21 are two efficient standards by MPEG [7, 21].

The Video Coding Expert Group (VCEG) is the group that developed a widely used series of video-telephony standards. H.261 and enhanced version H.263 and H.26L were the most famous standards.

The MPEG and VCEG groups collaborated on the Joint Video Team (JVT). This collaboration led to revising the final version of H.26L and converted it to be H.264/MPEG-4 Part 10. The last version became an international standard published by ISO/IEC and ITU-T [7, 11].

2.2.1.3 H.264 Codec Structure

The major phases in H.264 CODEC are similar to the prior versions in prediction, transformation, quantization and entropy coding. Enhancements added to the current version are in the details of each phase block [11, 14].

H.264 presents three main profiles: Extended, Baseline and Main. Each one contains a set of functions and coding steps in order to be most suitable for particular coding cases. These profiles contain different types of slices where slices are sequences of macroblocks which are processed in the order of a raster scan. The picture maybe split into one or several slices and a picture in H.264/AVC is a collection of one or more slices [7, 11].

Slices are self contained in the sense of the active sequence and picture parameter sets. Some information from other slices maybe needed to apply the deblocking filter across slice boundaries. The first three coding types are very similar to types in previous standards with the exception of the use of reference pictures [7, 9, 14].

H.264 has 5 types of slices; each slice can be coded using different coding types as follows:

I slice: All macroblocks of this slice are coded using intra prediction.

P slice: All macroblocks here are predicted using the motion compensation prediction (MCP) with a single list called list 0. The number of reference frames for list 0 is 5 frames. The current frame here is predicted only from the previous or the next frames [7].

B slice: Macroblocks of the B slice can also be coded using (MCP) but from two lists, list 0 and list 1.The number of reference frames for list 0 is 5 frames and 5 frames for list 1. The current frame here is predicted either from the previous frames or the next frames [7, 11].

SP slice: This is a so-called switching P slice that is coded such that efficient switching between different pre-coded pictures becomes possible.

SI slice: This is a switching I slice which allows an exact match of a macroblock in an SP slice for random access and error recovery purposes.

Slices as defined above may contain one or more macroblocks. Macroblock is a term used in video and image compression. The macroblock size was fixed to 8x8 pixels in the earlier codecs. In the H.264 video codec the macroblock size varies from 16x16 down to 4x4 pixels.

The H.264 codec adopts several macroblock sizes, the large macroblocks with 16x16, 16x8, 8x16 and 8x8. These sizes in some places are called Blocks. The small size macroblocks are P8x8 which contains: 8x8, 8x4, 4x8 and 4x4, these sometimes are called sub-blocks.

The main idea behind dividing macroblocks into smaller sub-blocks is to compress more precisely by selecting the most suitable mode for each sub-block rather than choosing the same mode for the entire macroblock. All sub-blocks should be combined to form 16x16 after compression and the cost is calculated to find the total sub-blocks cost for the 16x16 macroblock to compare it with other large 16x16 macroblocks.

The H.264 encoder contains many steps to achieve an efficient compression, Figure. Motion estimation (ME) and motion compensation (MC) are used for inter coding. T is transformation and Q is quantization. Inter and intra are the coder prediction methods.

There are three main coding modes in H.264 to code a given macroblock: The first one is SKIP mode which refers to the 16x16 mode where neither motion nor residual information is encoded and no motion search is required. Therefore, it has the lowest complexity in the mode decision process [7, 21].

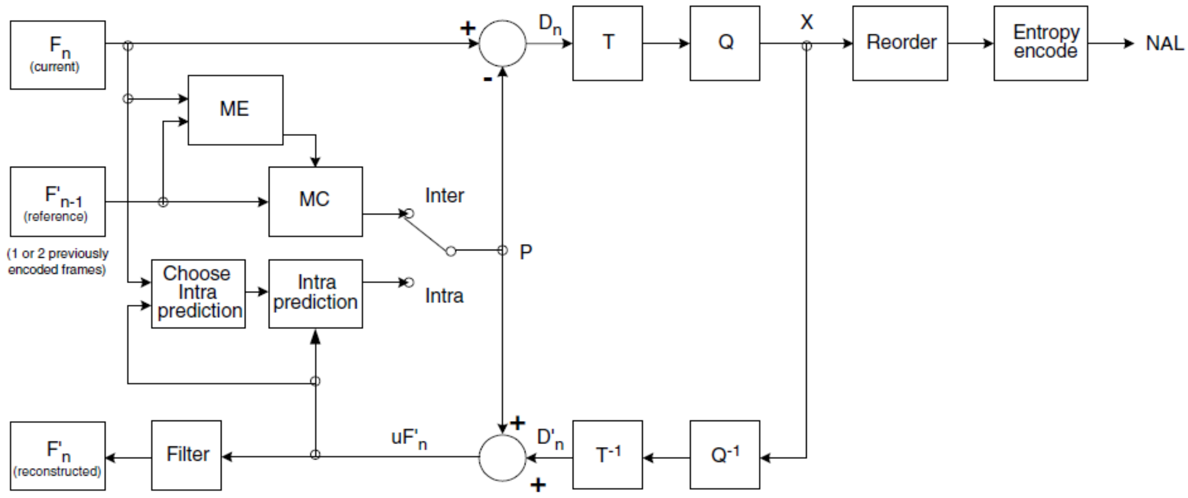

Figure 2-2 H.264 Encoder [1]

The second mode is Intra prediction which refers to encoding video frames without temporal prediction. Encoding samples in these Intra modes depend on the neighbor samples for the current macroblock. Samples are predicted using its neighbors in the same frame. In H.264/AVC, two different types of intra prediction are possible for the prediction of the luminance component Y [7, 21].

The first type is called INTRA4×4 and the second one is INTRA16×16. Using INTRA4×4 type, blocks of size 16 by 16 elements (16×16) are divided into sixteen 4×4 macroblocks and a prediction for each 4×4 macroblock of the luminance signal is applied individually. For the purpose of prediction, nine different prediction modes are supported.

One mode is DC prediction mode, where all samples of the current 4×4macroblocks are predicted by the mean of all samples neighboring to the left and to the top of the current block which have already been reconstructed at the encoder and at the decoder side.

In addition to DC-prediction mode, eight prediction modes with a specific prediction direction for each mode are supported: Mode 0 (vertical prediction) and Mode 1 (horizontal prediction). For example, if the vertical prediction mode is applied all samples below sample A are predicted by sample A. Also, all samples below sample B are predicted by sample B and so on. See Figure 2-3.

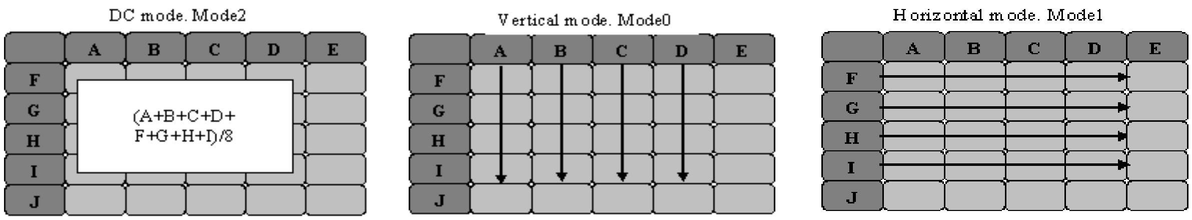

Figure 2-3 (DC, Vertical, Horizontal) Intra Prediction Modes

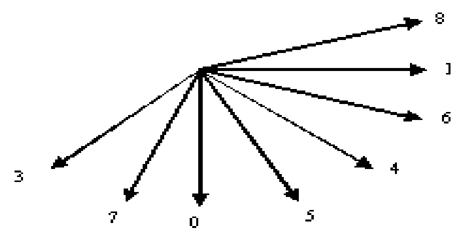

Figure 2-4 The Possible eight intra 4x4 prediction modes

The remaining possible directions for 4x4 intra modes are shown in Figure 2-4. Mode 0 and 1 represent vertical and horizontal modes simultaneously. Modes 3-8 are the other possible prediction directions. See Figure 2-4.

Using INTRA16x16 intra modes, one of the four prediction modes is selected for the whole block. The four prediction modes that are supported for INTRA16x16 are: vertical prediction, horizontal prediction, plane-prediction and DC-prediction [7, 15].

Plane prediction produces a plane with luminance gradient from light, the upper-left side, to dark, the lower-right. Hereby, plane-prediction uses a linear function between the neighboring samples to the left and to the top in order to predict the current samples. This mode works very well in areas of gently changing luminance [11, 15].

Using these modes is the same as with 4×4 prediction modes. Only the difference is that they are applied for the whole Block instead of implementing for each 4x4 macroblock.

Intra prediction for the chrominance signals Cb and Cr of a macroblock is similar to the INTRA16x16 type for the luminance signal. This is because the chrominance signals are very smooth in most cases. It is always performed on 8x8 blocks using horizontal prediction, vertical prediction, DC-prediction, plane-prediction [14].

The third mode is the Motion Compensated Prediction. The case here is different. The macroblocks are predicted from an image signal for already transmitted reference images [7, 12].

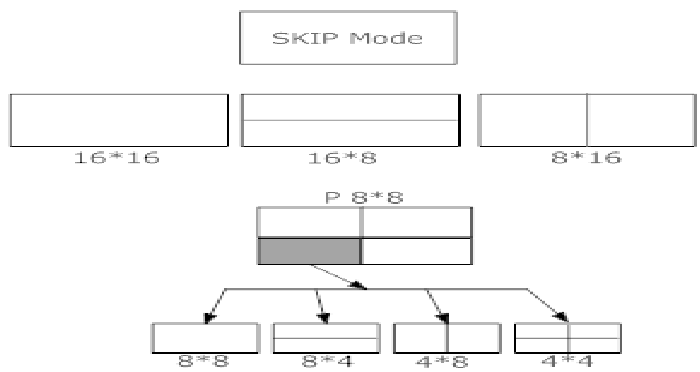

Figure 2-5 H.264 Inter Modes

Motion compensated prediction uses 7 modes for each block. Each block can be divided into smaller partitions with luminance block sizes of 16×16, 16×8, 8×16, and 8×8samples In case of 8×8 sub-macroblock in a P-slice, one additional possibility specifies if the corresponding 8×8 sub-macroblock is further divided into partitions with block sizes of 8×4, 4×8 or 4×4. See Figure 2-5.

Using inter modes in H.264 requires two major steps: motion estimation and motion compensation. Motion estimation is responsible for finding 16x16 block in one of the reference frames. The desired block is the closest match to the current block.

All blocks in inter modes are predicted from reference frames that are previously encoded by the codec. Reference frames may be prior or next of current frame in sequence display order.

After best match is selected by the step of motion estimation, the motion compensation step starts its process. It subtracts the previous macroblock from the current one. The result of subtraction is called residual macroblock. Residual macroblock contains luminance and chrominance data. This information with motion vector are encoded and transmitted [7, 11].

In the Encoder phase, the residual is encoded and decoded to reconstruct the macroblock and to use it as a reference for future motion compensation predictions.

The motion compensation block size is an important factor. As mentioned before, motion compensation uses seven different sizes for macroblocks. Using small blocks gives smaller residuals and thus smaller motion compensation results are achieved [13].

On the other hand, using such small blocks increases the encoder complexity. The complexity is affected by increasing the search operations needed. Motion vectors are increased too, and this increases the amount of data to be transmitted [22].

2.2.1.4 Rate Distortion Optimization

The H.264 codec has nine modes for intra 4x4; four modes for intra16x16 and seven modes for inter coding. In order to determine the best mode for a given macroblock, a cost function which includes both the distortion and the bit rate is used. The aim is finding the mode which has minimum cost [7, 14, 23].

The Lagrange multiplier optimization technique provides a systematic way to select the optimal coding mode. It uses a Lagrange cost function as in (2-5).

$$J = D + \lambda * \mathrm{R} \qquad (2-5)$$

where:

D is the distortion for the macroblock.

R is the number of bits used to encode the macroblock.

λ is a real number called a Lagrangian multiplier which depends on the quantization parameter (QP).

If we denote a sub-block by s, then the Lagrangian cost is also calculated from the sub-blocks as in (2-6).

$$J = \sum_{s=1}^{n} (D_s + \lambda * \mathrm{R}_s) \qquad (2-6)$$

where:

n is the number of sub-blocks in one macroblock.

$\mathbf{D_s}$ is the distortion for the sub-blocks.

$\mathbf{R_s}$ is the number of bits used to encode sub-blocks.

(2-4) and (2-5)are equal. The main difference is that the second one is used to calculate the Lagrangian cost for sub-blocks if small modes are being used, small modes means modes less than 16x16.

The distortion **D** is measured by one of the equations, mean squared error (MSE), mean absolute error (MAE) or sum of absolute error (SAE). See (2-7), (2-8) and (2-9).

$$MSE = \frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \left(C_{ij} - R_{ij}\right)^2 \qquad (2-7)$$

$$MAE = \frac{1}{N^2}\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\left|C_{ij} - R_{ij}\right| \qquad (2-8)$$

$$SAE = \sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\left|C_{ij} - R_{ij}\right| \qquad (2-9)$$

where:

N x N is the number of samples for the motion compensation block size.

C_{ij} is the current area samples.

R_{ij} is the reference area samples.

These equations represent the difference between the original signal block **s**, and the corresponding reconstructed signal for the same macroblock. The rate $\mathbf{R_x}$ is the rate that is required to encode the block with the entropy coder [7, 11].

The Full Search algorithm is used to examine all possible prediction modes to find the best one. We can divide the algorithm into two phases:

Phase one:

The major steps for intra prediction mode selection are as follows:

Step1: Find the Best Intra 16x16 Prediction Modes.

In intra mode, a block is predicted based on previously encoded and decoded blocks and is subtracted from the current block.

• Generate four prediction blocks according to the four intral6x16 modes.

• Calculate the sum of absolute error (SAE) between the original and the predicted block. The sum of absolute error (SAE) is used because of its computational simplicity.

• Compute the cost of the block.

• Find and choose the mode with minimum cost.

Step2: Find the Best Intra 4x4 Prediction Modes.

Divide the macroblock into sixteen 4x4 non-overlapped blocks. For each 4x4 block find the best mode as follows:

• Generate nine 4x4 blocks based on the nine Intra 4x4.

• Calculate the sum of absolute error (SAE) between the original and the predicted block.

• Compute the cost of the block.

• Find and choose the mode with minimum cost.

Step3: Find the Best Intra Prediction Mode.

If the cost from step (1) is larger than the cost from step (2), then intra 4x4 is selected, otherwise 16x16 is selected [7, 24].

Phase two:

The major steps for the inter prediction modes selection process used in the reference software which can be summarized as:

• Perform motion estimation for each 16x16, 16x8 and 8x16 size in a macro block

• Encode the macroblock with the different sizes and compute the costs. Here if 16x16 macroblock is chosen then the RDO is calculated once for the entire macroblock, if the 16x8 or the 8x16 macroblock is chosen then the cost will be calculated for two 16x8 or 8x16 macroblocks respectively.

• For each 8x8 block in a macro block, perform motion estimation for each 8x8, 8x4, 4x8 and 4x4 sizes. Here if 8x8 macroblock is chosen then the RDO is calculated two times for the two 8x8 macroblocks, if the 4x8 or the 8x4 macroblock size is chosen then the cost will be calculated for eight 16x8 or 8x16 macroblocks respectively.

Finally if 4*4 block size is chosen, then the cost will be calculated for sixteen 4*4 sub-blocks.

• Choose the mode that gives the minimum cost among 16x16, 16x8, 8x16 and P8x8 which contains 8x8, 8x4, 4x8 and 4x4 sub-blocks.

Phase of mode decision:

Compare result from phase one to the result from phase two, then choose the minimum**.**

When is it better to choose intra prediction or inter prediction?

As mentioned above, the intra prediction is predicted from the current frame which is for I-type frames, while for inter prediction the prediction is done using the previous frames which is for P-type and B-type frames.

The inter prediction is more suitable than the intra modes for high quality video sequences, which usually have a high number of frames, which means a high frame rate [15, 25].

There is a high chance to find two successive frames which are very similar, and this is the most important goal for inter prediction; it uses the temporal redundancy to reduce data needed to encode the current frame.

2.2.2 MPEG-4

MPEG stands for Motion Picture Expert Group which is a group of experts established by ISO (International Organization for Standardization) and IEC (International Electro technical Commission). MPEG has devised many standards for video and audio compression and transmission technologies [7].

The MPEG standard includes different codecs and versions. MPEG has been adopted by many companies as it is built to cover different classes of video applications.

The MPEG-4 has the capability to enable developers to create better multimedia objects and improvements of quality in several services such as digital television, animations and the internet. MPEG-4 has many features which make it one of the most practical codecs [7].

The feature of fitting 2-D and 3-D animation graphics is one important advantage of the codec. In addition, it supports network application of real transmission techniques, high error resilience and coding the foreground of images separately [26].

MPEG-4 provides a set of profiles and encoding tools. Subsets of the tools are organized to use with specific types of application in a so-called profile. The MPEG-4 profiles can be summarized in five types as follows [26]:

- Simple profile: used with rectangular frames.
- Advanced core profile: used for still texture and video objects.
- Still texture profile: used with still texture.
- Object based profile: used with random shapes.
- Studio profile: focuses on visual quality and compression efficiency upgrading, this profile is used inside the studios.

The MPEG-4 has a high ability to deal with video objects and animation. Thus this research will study this codec to evaluate its compression efficiency of video graphics compared to some other codecs.

2.2.3 VP-8

VP-8 codec was created by On2 Technology which was bought by Google. VP-8 was developed to achieve an outperformance of H.264. The goal was achieved whereby VP-8 outperformed H.264 in several services.

VP-8 achieves high compression quality at low bit rate. In addition this achievement is done with lower compression complexity. These features led to consider VP-8 as a strong rival to H.264.

The libvpx is the only software library that is able to run VP-8 encoder with two frame structures, I and P frames. The B frame is replaced by a so-called technique Golden frame. VP-8 codec will be evaluated in this research and the compression quality of video graphics will be compared to the H.264 and MPEG-4.

2.3 COMPRESSION OF GAMES VIDEO

2.3.1 Screen Resolution

Screen resolution and picture size are terms which refer to the number of pixels in an image or video frame. Number of pixels is computed by multiplying the horizontal pixel size by the vertical one. An image storage size and resolution increases when number of pixels increases.

The resolution selection depends on the need of the image or video frame. While some applications require high quality images such as medical services, other applications are available to deal with lower image and video frame resolutions, such as video conferencing and low bandwidth Internet streaming [7, 27].

Many standards of image resolutions are available; these standards are organized according to the applications that use them. The main resolution standards are: Computer graphics, Television, Film and Video conferencing [7].

As this research is interested in video graphics and video gaming, the resolution standard of computer graphics is only considered because it is suitable for the small screen resolutions.

Standard	Resolution	Pixels
SQCIF	128x96	12288
CIF	352x288	101376
4CIF	704x576	405504
QCIF	176 x 144	25344
16CIF	1408 x 1152	1622016
QVGA	320 x 240	76800

Table 2-1 Screen resolution standards

The video conferencing standard screen resolution consists of six major image sizes. These sizes are shown in Table 2-1. This research will use three of these frame resolutions QCIF, CIF and 4CIF as they are the most popular resolutions in video graphics [27].

2.3.2 Frame Rate

A moving video involves successive frames shown at periodic intervals of time. This moving video is produced by capturing snapshots successively. Playing the captured series of frames gives the video motion. Temporal sampling or frame rate affects the smoothness of video motion as higher frame rates give higher motion smoothness. However, this requires more samples which means higher file size [7].

Frame rate refers to the number of frames displayed in a period of time, frame rates below 10 frames per second (fps) are used for low communications as a low amount of data is required, however, these low frame rates produce jerky video scenes. The bit rate between 25-30 fps is the television standard which gives moderate smoothness with an acceptable data rate [7].

This research will consider several bit rates and the selected bit rates are the typical uplink rates used at the present time.

2.4 ON-LINE GAME SERVICE PROVIDERS

Xfire is one of the most well-known game server browsers among on-line game users [2]. Xfire switched from the LiveStream [28] to the TwitchTV [1] platform in 2012. TwitchTV allows users to broadcast live video streams of their games while playing. The user needs Xfire client to connect his own PC to a server and then detects other advertised games. The new gamer is shown to other players as a friend. Xfire also provides services to show which server players are playing on, the level that is running and the ping times. LiveStream known as "Mogulus", TwitchTV and JustInTV [1, 28, 29] are examples of platforms that stream the game that the user is playing. However, there are many game server browsers that provide interfaces to utilize gamers chatting, allow checking game progress, creating gamer groups and providing multiple chatting services [30-36]. These server browsers don't utilize broadcasting. There are many examples of these service utilities such as:

1. GameRanger (Windows and Mac).
2. GameSpy Arcade.
3. Garena.
4. HLSW.
5. Kali.
6. Steam.
7. XQF (for UNIX/Linux).

3 RELATED WORK

3.1 INTRODUCTION

The H.264 codec has attained more attention the last few years. Many studies in the literature have focused on the H.264 codec and its specifications. Much of this research spotlighted the complexity reduction and computational time saving. A thorough review of complexity reduction techniques of H.264 will be described in this section. The section will focus on the recent proposed fast encoding techniques for H.264 video coding.

Most studies on video compression have been focused on real video sequences and video codecs' ability to compress this type of sequences. Even though the on-line gaming field depends on video compression, it still lacks research. The study of recommending a tradeoff between frame rates and resolution within the typical uplink streaming is also considered a new research field. However, brief descriptions of the available synthetic video compression studies are given in this chapter.

Category Name
Early termination using SKIP mode
Dividing modes depending on their characteristics
Modifying and replacing some H.264 mathematical steps to speed up the encoder
Control the reference frames checking mechanism
Other local optimization methods

Table 3-1 Local Optimization Approaches Categories

3.2 FAST ENCODING TECHNIQUES for H.264

The proposed fast encoding techniques for H.264 codec can be categorized in five main categories as shown in Table 3-1. Each category contains the most similar approaches. The detailed descriptions are stated in the following sections.

3.2.1 Early Termination Using SKIP Mode

The SKIP mode is the simplest mode to be selected. This mode is encoded directly without considering the encoding of motion or residual information. Avoiding using motion search decreases the complexity. Therefore, the SKIP mode has the lowest complexity in the mode decision process [7, 14, 37].

As a result of this basic idea, much research has focused on the SKIP mode features to save some of the encoder complexity time. The main idea was achieved by testing the SKIP mode directly at the first stage. If the result met some specific criteria, the SKIP mode comes to be considered as the best mode and the process will be terminated.

In [38], the mode search algorithm is divided into two phases. The first phase is the SKIP mode phase. The block is processed as SKIP mode. Then a set of conditions is implemented to guarantee the suitability. Once all conditions are met, the SKIP mode is chosen and the process is terminated. The second phase functions are implemented if the SKIP mode was not adequate depending on the first phase conditions.

In [39] an early termination method depending on the SKIP mode is presented. The SKIP mode cost calculates with the 16x16 inter mode. A comparison between both costs is made in order to choose the minimum one. In a case where SKIP mode is better, it is chosen as the best mode and the process terminates. The process continues if SKIP mode is not better.

The algorithm continues dividing into smaller macroblocks and comparing to find the best, once larger macroblocks result in the lesser cost, the search is terminated. This means termination happens once the larger block has better cost than the smaller one.

The method in [40] classifies the modes into several categories depending on the motion activity that could be processed by each mode. Two thresholds are specified in advance, high and low threshold. The SKIP mode is selected if its cost is less than the lower threshold, and the process terminates. The other groups of modes are applied if SKIP mode cost is not less than the lower threshold. This is determined depending on the value of SKIP mode cost.

In [41], there is the proposal to check the SKIP mode first. The SKIP mode is chosen when motion vectors all are zeros. Once SKIP mode is chosen as the best mode the process is terminated. When SKIP mode is not chosen, the algorithm proceeds to ordinary testing of inter modes cost and compares them by a pre-calculated threshold. If any mode gives less cost than the threshold, the process terminates and the mode is chosen to be the best mode.

In [42], several conditions for P-slice and B-slice are specified in advance. In case of SKIP mode meeting these conditions, it is chosen as the best mode and the process terminates other than for the remaining modes that need to be checked.

A limitation of early SKIP mode techniques is the number of omitted modes. In cases where SKIP mode meets the condition of terminating the mode checking process, this leads to skip high possibilities of finding a better mode between inter and intra modes. Also, in order to check the SKIP mode, the reference frame is just one previous frame. Thus, early termination SKIP mode also misses the benefit of searching multiple references which decreases accuracy.

In [43], an early termination technique based on the use of an all-zero-block (AZB)detection algorithm is proposed. AZB is presented in [44] and exploited in this paper. It involves the idea of empirical analysis of the Inter-layer correlation of AZB.

In [45], an early termination technique according to computed mean error and standard deviation is presented. The modes are put in groups and a group is checked if the values of mean error and standard deviation have met a determined range. Once the chosen group of blocks is checked, there is no further need to check other modes and the process terminates.

In [46], a list of the most priority-based mode is constructed based on utilizing the coding information of temporal and spatial blocks. The list is sorted in descending order and process terminates once a mode met specified criteria.

In [47], an intra early termination technique is proposed. Thresholds are determined in advance, and the suitable mode is chosen depending on a fast computational value called SAHTD. The algorithm calculates the SAHTD value and determines which thresholds it is within, and then a mode is chosen accordingly.

In [48], a SKIP mode detection approach is proposed for efficient SKIP mode complexity reduction. The idea uses the coding information of spatial neighboring MBs and co-located MB in base layer to predict the SKIP mode. If the coding information meets SKIP conditions, intra prediction and ME variable size is skipped.

3.2.2 Dividing Modes Depending on their Characteristics

Many researchers have tried to speed up the encoder using modes characteristics. Studying the nature of each mode gives frequent opportunities to choose it directly. The encoder checks the most probable mode first.

The idea of mode classifying with early termination can be performed in several approaches. One approach involves pre-studying the mode's specifications, and determines the best mode for the current block depending on the block characteristics [49, 50].

Another approach focuses on building a priority queue for all provided modes [51]. The method updates the priority queue of modes while the encoding process is running. This helps the encoder to check the highest priority first, and this leads to time saving.

One more approach divides modes into several groups; each group containing the most similar modes. The approach is conducted either by selecting a mode from each group to be evaluated, then the group with the best mode is selected to be checked, or the first mode of first group is evaluated by a pre-defined threshold [52].

In [53], all possible modes are divided into four groups. One mode from each group is checked, the lowest complexity mode indicates the best candidate group where it was found. Only modes in the best group are checked and their complexity is calculated.

In [54], a statistical study recommends that subblocks of inter 8x8 and intra 4x4 modes cost the most computation time of the total encoder time. Thus, modes are divided into inter and intra modes, where inter modes are checked from large modes down to smaller ones. When larger modes give less cost than smaller ones, they are chosen as best mode. The intra modes are checked in a similar way. If a larger mode is better than a smaller one, the process terminates and the larger is judged to be the best intra mode.

In [55], a method of using some features is defined. The features are defined in terms of temporal and spatial correlation representation to categorize all modes. Five features are defined to implement mode's classification, three features for temporal and two for spatial correlations. Mode classes are classified hierarchically according to these features.

In [56], two main categories are defined, the high probability macroblocks category containing inter 16x16 and inter 8x8 and a low probability category containing the remaining modes.

Mode's classification depends mainly on the so-called motion classifier and probability classifier. Each mode should be tested by the probability classifier to determine if it is high or low probability. In high probability cases, the rate distortion cost is tested by the motion classifier for further division.

In [57], a statistical learning model is proposed. A pre-classification approach is presented to extract features and determine the best results from these features. The features are best block size and number of reference frames.

In [58], a multi-phase classification scheme model is proposed. The model builds a mode decision tree according to the clustering of rate-distortion costs. In the model, the nearest cluster means measure is used to examine candidate modes phase by phase. A performance control mechanism is incorporated to maintain coding performance.

In [59], an early termination model of the Inter-prediction mode decision is proposed. The model is designed for hierarchical B pictures' temporal scalability. The early termination model is tested in two sets of hypotheses: mean and variance so as to check the similarity between pixel values of current block and others of a reference block of sizes 16X16 and 8X8. The confidence intervals are adjusted in temporal scalability levels considering the ratio of large block mode selection in hierarchical B picture GOP structures. The proposed algorithm provides an effective compromise between encoding time saving and coding efficiency.

3.2.3 Modifying and Replacing some Codec Mathematical Steps

The exhaustive mode decision adopted by JM reference software takes about 90% of the total computational time of the H.264 encoder when CABAC entropy coding is used[40]. Thus, several fast encoding schemes have been proposed based on modifying and replacing some encoder computations to speedup the encoding process.

In [60, 61], a new cost calculation for inter mode selection is proposed. Two new methods were developed to replace former high computation calculations. The entropy coding was replaced by a so-called Bit Rate Estimation. This method is less computationally expensive than the original entropy coding and gives similar results. Transform Domain Distortion (TDD) is added to the encoder phase. By the TDD measuring technique, older methods of pixel reconstruction and inverse transformation are eliminated and replaced by this new method.

In [62], another technique of modifying the encoder mathematical steps is proposed. According to reference software, it uses two methods to select the best mode. The first is RDO and the second is sum of absolute transformed differences (SATD). The SATD technique is a very fast method but it causes more loss in PSNR than RDO. The authors proposed to use SATD as an early step. After this first step, all modes are determined and investigated for their effect on efficiency. The proposed algorithm decides to ignore the less probable modes and use the high probability modes.

In [63], a proposed idea to reduce coding process computations is presented. The idea eliminates inverse quantization, inverse DCT and reconstruction processes. The proposed algorithm estimates the distortion using coefficients calculated in the quantization process. The distortion is calculated using integer operations which lead to more hardware implementation efficiency.

In [64], an algorithm model with a new complexity rate distortion model to perform optimal complexity allocation among encoding tools is proposed. The model precisely describes how each encoding tool influences the complexity rate distortion performance of the encoder with brief formulas. The algorithm obtains the optimal complexity of each encoding tool by a closed-form solution with small complexity overhead.

3.2.4 Control the Reference Frames Checking Mechanism

In [65], an early termination process according to reference frames checking is proposed. The 16x16 macroblock is divided into four 8x8 blocks. The motion search is performed for the four blocks on the immediately previous frame. Therefore, the four values of motion vectors and corresponding rate distortion are obtained. If the values are not greater than a predefined threshold, the macroblock is highly probable to be a motionless part. In this case, the previous frame is selected as valid reference frame and the encoder process terminates.

In [66], the correlation of temporal complexity between a macroblock and the reference region in the previous reference frames is studied. It proceeds according to the assumption that increasing temporal distance between the current macroblock and the reference frame increases the temporal complexity. The temporal complexity between the current macroblock and the closest reference frame is also highly correlated. The method studies the correlation of the current macroblock with the closest reference. If it meets the region conditions of the current macroblock the first reference frame is valid and no other references need to be checked. One more reference is checked each time the reference frame has not met region conditions.

In [67], a priority reference frame selection technique is proposed to reduce the number of reference frames to be checked. The priorities of all reference frames are computed using spatial and temporal correlation of the reference frame index and motion vectors. Then, the candidate reference frame is selected to reduce reference frames for each motion estimation block.

3.2.5 Other Methods

In [68], a data mining algorithm to develop a decision tree for the encoder mode decision is proposed. The complexity reduction is conducted by building a binary tree and searching the suitable leaf depending on certain metrics. This method uses techniques of data mining to take advantage of the correlation between the macroblock statistics and macroblock coding modes. A tool was used to build up a decision tree for H.264 coding mode decisions. The proposed algorithm only requires the calculation of macroblock statistics such as variances and differences for the border pixels.

In [6], an idea of motion homogeneity is proposed. This algorithm begins by performing motion estimation on a 4x4 block size. An equation is generated to calculate the motion homogeneity direction. Results of the equation are compared with a threshold using equation metrics. After that, the mode is determined as to which class between five classes it belongs to. According to the algorithm, only modes in the selected class will be tested by RDO. The modes are classified depending on pre-defined statistical study.

In [69], a process of checking the correlation between the current mode and the previous mode is proposed. If a mode correlation is less than a predefined threshold, the mode is chosen as best mode and there is no need to check the remaining modes. The algorithm developed equations to define the spatial and temporal mode prediction.

The threshold is calculated by multiplying a constant by cost of the reference block. The cost for candidate modes is calculated by assuming the previous block frame in temporal correlation and the neighbors for spatial correlation.

In [70], a simplified mode selection model is proposed. The model is based mainly on a new Lagrangian cost function technique using distortion, number of motion vector bits and block type information.

In [71], an algorithm of block size partitioning prediction for MPEG-2 to H.264 transcoding is presented. The algorithm estimates block size partitioning using rate distortion optimization techniques with predicted initial motion vectors. To reduce transcoding complexity, small block sizes are avoided in transcoding.

In [72], a parameterized complexity scheme is proposed. The scheme trades off complexity and coding performance to meet real-time and power constraints.

In [73], a statistical analysis of motion estimation factor keys is prepared. According to this statistical study, the validity of fraction motion estimation reduction technique criteria is proved. Any MB satisfies this motion estimation criteria can keep its integer motion estimation. The idea avoids less probable but high cost prediction modes.

In [74], a scalable model of cogeneration of fast motion estimation processors and algorithms is proposed. The paper presents a motion estimation processor capable of supporting the processing requirements for high-definition video using H.264. The report concludes that it is possible to match the processing requirements of the selected motion estimation algorithm to the hardware micro architecture that leads to an efficient implementation.

In [75], the usage distribution of intra prediction is investigated. The prediction modes for the focus region of the picture being coded are then applied. For the other regions, it utilizes only the important modes where important modes are determined from the usage in the previously coded frames.

In [76], motion compensation of the enhancement layer is performed in both the same resolution layer and in the corresponding lower layer in the SVC extension of H.264. Therefore a fast Inter-frame and Inter-layer mode selection algorithm to exploit the motion activity in the key frames of the video sequence is proposed. A statistical study was conducted and obtained a result showing that a slow movement macroblock is more likely to be best matched by one in the same resolution layer. For macroblocks with fast movement, motion estimation between layers is required.

3.3 CODECS COMPARISON and TRADING-OFF STUDY for SYNTHETIC VIDEOS

The study of synthetic videos is considered a new field of research. Many studies have been done on video codec comparison. But after much searching, no previous research has been located on the video codecs capability for synthetic videos compression. In addition, the study of computer graphics sequences and attempting to recommend best tradeoff between frame rate and frame resolution within typical uplink speeds also lacks research. In fact, this type of research is distinctive and there has been very little previous research on this area.

In [77], a study of the effects of frame rate and resolution on user performance for computer game is reported. The First Person Shooter (FPS) game was used in the study. The study concludes that frame rate has a significant impact on player performance and enjoyment.

In [78], effects of resolution on user playing FPS are presented. The effect of resolution in high and low contrast in addition to full screen and windowed are considered in the study.

In [79], a study of frame rate and resolution effects on the most common game actions, such as, shooting and navigating is reported. The study concludes that frame rate has a much greater impact than frame resolution.

In [80], a study of the relative importance of spatial resolution, temporal resolution and "intensity" resolution is presented. The study concludes that the interactive task is highly affected by the temporal resolution, whereas spatial resolution is much less important.

In [81], a comparison study between H.264 and Motion JPEG2000 for high definition video coding is described.

In [82], the performance of H.264, MPEG-4, H.263 and MPEG-2 is considered. The codecs are compared using PSNR and subjective testing schemes.

In [83], an evaluation study of H.264 performance is reported. The trade-off between coding efficiency and error resilience for network applications is considered. In addition, the end-to-end delay is considered in this evaluation study.

In [19], an evaluation of perceptual visual quality under various settings and requirements is described. The subjective assessment tests are analyzed to study the influence of the different dimensions on the subjective evaluation. The dimensions are: encoder type, video content, bit rate, frame size and frame rate, where H.264 and H.263 are the codecs used in the study.

In [84], a methodology to evaluate the perceived video quality when watching high motion videos is reported. The video sequences are chosen from some football matches with different physical quality metrics. CIF and QCIF frame resolutions are used to study the relationship between frame rate and quantization and its impact on the perceived quality.

In [85], a new objective quality metric for video sequences is proposed. The proposed metric is based on estimating the MOS of compressed sequences using MSE, spatial video content and skin information. The study is focused on the factors that attract viewer attention.

In [86], a review of video quality measurement techniques is presented. Both subjective and objective quality metrics are discussed. Also, a hybrid metric is considered in the study.

In [16], an evaluation study of present video compression techniques effects on the perceptual quality of video coding is presented.

In [87], a study of the rating scale in subjective testing methodology is presented. The study focuses on the importance of the rating scale elements and the effect on the subjective evaluation results. Four different rating scales are used by participants. An analysis of the rating results is conducted to evaluate the different proposed rating scales.

4 COMPARISON OF COMPRESSION TECHNIQUES FOR SYNTHETIC VIDEO

4.1 INTRODUCTION

The fast growth of communication and computer networking has given more flexibility to the application providers to increase the service quality of their products. The on-line gaming producers constitute one sector that has developed its applications to exploit this development. As a result, newer on-line games consume more bandwidth and require higher bitrates. Therefore, video up-to-date compression techniques are required to reduce the bit rate to appropriate levels without compromising quality.

The compression of synthetic videos and computer graphics is a new field in video compression. According to the literature survey that was conducted, there has been no published work on this area. Therefore, a general comparison study between the most popular video compression techniques and their suitability for computer graphic compression is presented below.

Three types of video codecs are selected for this comparison study: H.264, MPEG-4 and VP8. The first two codecs are preferred in current media communication since both produce high quality results with various bit-rates. VP8 is now considered an efficient codec and is comparable to H.264 in it is performance.

The sequences used to perform this comparison study have been captured from the game "World of Warcraft". This game is one of the most popular on-line games. It contains a very wide range of features and visual specifications. The sequences have been captured using an expert capturing software called FRAPS [88]. This professional and freely downloaded software was used to record video while the game is being played.

The comparative study of video codecs is described in detail in the next section, the choice of a video encryption scheme depends on the application context [89].

4.2 COMPARISON AND EVALUATION STUDY

The main idea of video compression for on-line game streaming is to maximize the amount of data transmitted in the network within the bandwidth limitations. The main benefit of a video codec is the ability to reduce video size while conserving quality and fidelity. There are several parameters that affect the process of video compression; mainly the frame rate, bit rate and frame resolution.

The game player needs time to interact with others by viewing their screens. This is achieved by adding extra screens for each member, which makes small resolution frames more desirable. Small frame resolutions with QCIF and CIF sizes were selected for making comparisons in this study. Three different frame rates; 15, 25 and 30, were selected to represent diverse video sequences that may be used in on-line gaming.

The compression in this study was applied over 256, 512 and 1024 kbps bit rates. These bit rates were selected to suit the average internet speeds available in most countries. According to [90-92], the selected bit rates match the average available internet speed in the UK and in Jordan where the study was prepared.

The compression experiments were applied over diverse types of captured video sequences in order to provide different scene contents. The scenes include sequences with high-texture high-motion, high-texture low-motion, low-texture high-motion and low-texture low-motion contents.

The compression process is affected by the content of the sequence. Therefore, selecting different contents will make the comparison more comprehensive. The codecs were evaluated under different sequence contents which reveal the codec behavior in these different modes.

The different video sequences with all parameters proposed above were tested using the three selected video codecs. The compression process was applied using the FFmpeg software [93] under the Linux Ubuntu operating system. FFmpeg is a professional program that was designed for audio and video compression and conversion. One of its main advantages is providing efficient supporting libraries for performing video compression using different codecs.

FFmpeg is recommended to run under Linux, so Linux Ubuntu11.4 operating system was installed and then FFmpeg with all needed libraries and facilities were installed under this operating system. After completing these necessary preparation steps, the compression methodology was applied and conducted using the Linux command line window.

Intel Core i3 with 4 GB RAM was used for compression. All codecs were tested using their simple and fast features and profiles to provide suitable compression environment of on-line streaming services. All codecs' parameters are shown in Table 4-1, Table 4-2 and Table 4-3.

Parameter Name	GOP	B Slice	CABAC	RDO	Sample Depth	QP
Value	IPPP	No	No	Yes	8 bits	24

Table 4-1 H.264 Parameters

Parameter Name	Coding Tool	B Frames	CABAC	RDO	Sample Depth	QP
Value	Inter Prediction	No	No	Yes	8 bits	24

Table 4-2 MPEG-4 Parameters

Parameter Name	Coding Tool	Alternative Reference	CABAC	RDO	Sample Depth	QP
Value	Inter Prediction	No	No	Yes	8 bits	24

Table 4-3 VP-8 Parameters

The most important step after compressing the video sequences is to evaluate and compare the codec's performance. The step of evaluation and comparison will take into consideration all tested video sequences with their different contents using several collection of parameters.

The evaluation process used in this study to judge the codecs' performance contains different subjective and objective metrics. These methods are discussed in detail in Section2.1.

PSNR was used as an objective metric since it is widely used by the video and image processing community. Mean squared error (MSE) and PSNR are popular because they are easy to understand and implement, and they are fast to compute [86].

The PSNR metric is a mathematical way to calculate the ratio of reconstructed quality and noise, and it is a logarithmic representation of MSE. PSNR was calculated using FFmpeg and was produced directly after a sequence compression was completed.

Subjective video quality measurements depend on the perceived quality of video sequences by observers' assessment. The DSIS [94] subjective assessment metric was used in this study. Four ten-second test video sequences were viewed by fifteen viewers. For completeness, different spatial and temporal resolutions were considered.

The subjective DSIS variant II which is suggested by ITU-R recommendation [94] was conducted. The original video sequence was displayed to the viewer first; and then the decompressed video sequence was displayed. The viewers could not score before this process was repeated. After that, the viewers give their impression on the compression quality, compared to the original sequences, using a five-level quality scale.

All viewers were BSc students with sufficient knowledge of English. The experiments were conducted in a computer lab on 19 inch Dell LCD monitors. The lab's windows were covered and it was lit with white lights. All experiments were performed under the same conditions.

The comparative study with results analysis is presented in this section. The next two subsections will compare the effect of frame rate and bit rate on CIF resolution, and then the QCIF resolution will be discussed.

4.2.1 CIF Format Resolution

All captured video sequences were compressed in CIF resolution using the codecs under evaluation. The sequences were then tested and the results are shown below.

4.2.1.1 High texture High motion

Video sequences with fast object motion and multiple contents: text, colors, objects, etc., are considered complex to video codecs. This type of video sequences has a strong impact on the codec compression efficiency since the codec needs to deal with spatial and temporal content.

As shown in Table 4-4, H.264 PSNR outperforms MPEG-4 and VP-8 at 1024 kbps. H.264 PSNR was more than 2dB better than the other two codecs and the PSNR difference decreased as the bit rate was decreased. H.264 still performed better than other codecs at 512 kbps, and the difference decreased as the frame rate was decreased at the same bit rate. MPEG-4 performed slightly better with 256 kbps than with the other frame rates.

Bit Rate (kbps)	Frame Rate (fps)	PSNR (dB)		
		H.264	MPEG-4	VP-8
256	15	34.01	35.39	32.17
	25	33.63	35.05	33.97
	30	33.64	35.02	33.84
512	15	37.76	37.55	37.38
	25	37.42	37.09	37.26
	30	37.52	37.06	37.15
1024	15	43.33	40.68	41.17
	25	43.05	40.35	40.88
	30	43.27	40.13	40.61

Table 4-4 CIF – High texture High motion sequences- PSNR

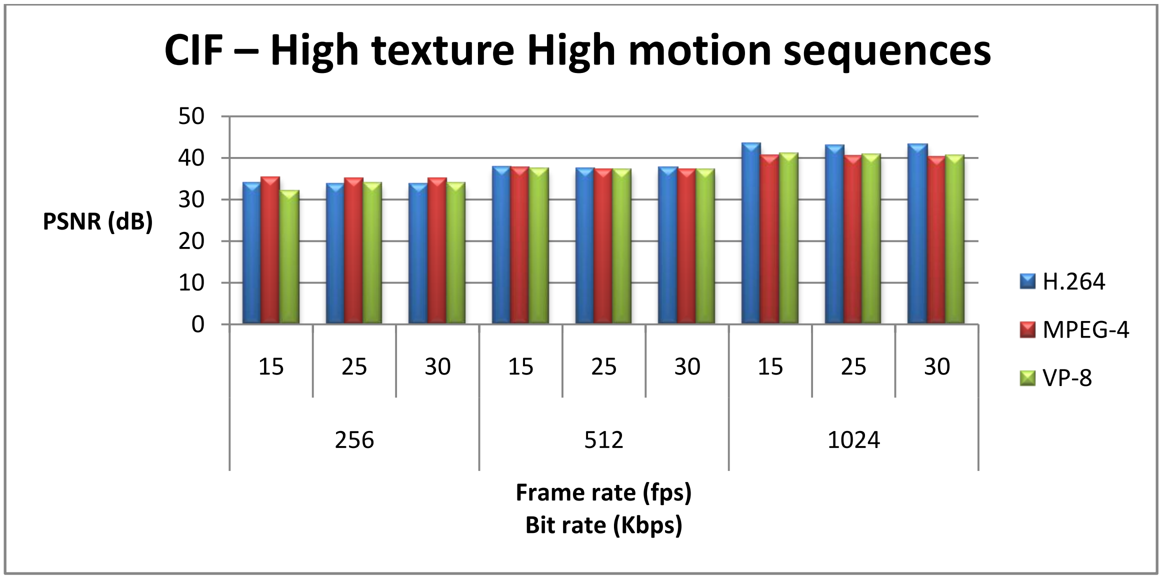

Figure 4-1 CIF – High texture High motion sequences– PSNR

Since this type of video sequences contains high texture and fast motion, the evaluation is easier and should be more accurate. The user can judge the outcome of the codecs according to the change of quality and the smoothness. On the other hand, the sequences' complexity with this huge content makes the compression more difficult. Increasing details and motion will respectively increase the coder complexity and compression quality.

Codec	Bit Rate		
	256	512	1024
H.264	4.2	4.33	4.46
MPEG-4	3.33	3.66	3.8
VP-8	3.73	3.93	4.26

Table 4-5 CIF – High texture High motion sequences - DSIS

The average DSIS is shown in Table 4-5. The results show that the users preferred the sequences compressed by the H.264 codec more than those compressed using MPEG-4 and VP-8.

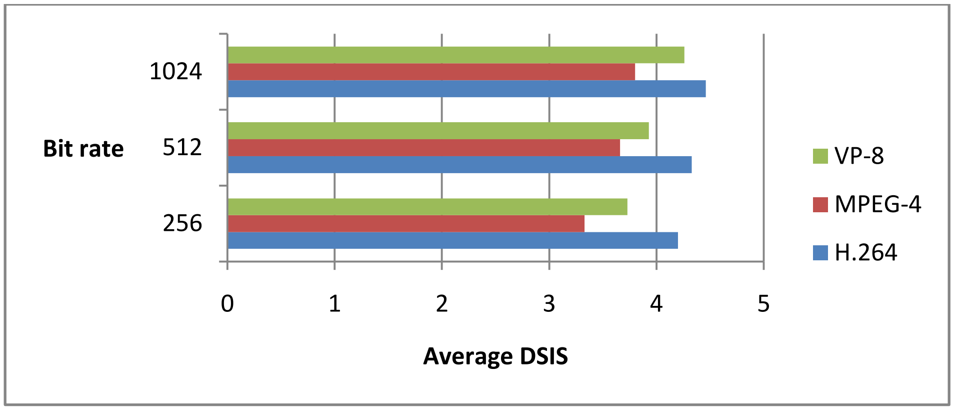

Figure 4-2 CIF – High texture High motion sequences– DSIS

By viewing the subjective and objective results of this video sequence type, H.264 outperforms other video codecs except for the objective result at the 256 kbps bit rate. At this bit rate, MPEG-4 performs better than other codecs. In addition, higher results of MOS are generally associated with higher resolution of spatial and temporal components.

In Figure 4-3, snapshots of the original and compressed video sequences at 1024 kbps are shown. The outcome of H.264 colors and shapes remained similar to the original. MPEG-4 and VP-8 kept the shapes but the images became rather blurry.

	
ORIGINAL	H.264
	
MPEG-4	VP-8

Figure 4-3 High texture High motion at 1024 kbps

In Figure 4-4 and Figure 4-5, it can be seen that the distortion increased as the bit rate was decreased. The distortion is very noticeable in VP-8 while there is less distortion in MPEG-4. Conversely, H.264 does not show a considerable big loss of quality for the lower bit rates.

	
ORIGINAL	H.264
	
MPEG-4	VP-8

Figure 4-4 High texture High motion at 512 kbps

	
ORIGINAL	H.264
	
MPEG-4	VP-8

Figure 4-5 High texture High motion at 256 kbps

4.2.1.2 High texture Low motion

Results in Table 4-6 demonstrate the average PSNR for the sequences of the three codecs. H.264 shows, on average, a 3 dB better performance than MPEG-4 and VP-8 at 1024 kbps. The difference between H.264 and VP-8 decreased at 512 kbps where it is less than 1 dB on average. The VP-8 codec shows slightly better performance than H.264 at 256 kbps.

Bit Rate (kbps)	Frame Rate (fps)	PSNR (dB)		
		H.264	MPEG-4	VP-8
256	15	42.6	39.31	43.01
	25	41.56	39.65	42.18
	30	40.95	39.63	41.11
512	15	47.96	42.52	46.38
	25	46.76	42.44	45.95
	30	46.09	42.37	45.42
1024	15	55.5	44.6	52.67
	25	53.26	44.77	50.06
	30	52.33	44.79	49.16

Table 4-6 CIF – High texture Low motion sequences – PSNR

As illustrated in Figure 4-6, VP-8 slightly outperformed other codecs at lower bitrates. On the contrary, H.264 shows larger PSNR differences at higher bit rates.

VP-8 codec shows a slightly better performance than H.264 at the low bit rate. However, this slight was significant if the low VP-8 compression complexity was also considered since the H.264 codec normally consumes more compression time compared to VP-8. As a result, the H.264 codec performed better at higher bit rates and it is recommended for such types of video sequences.

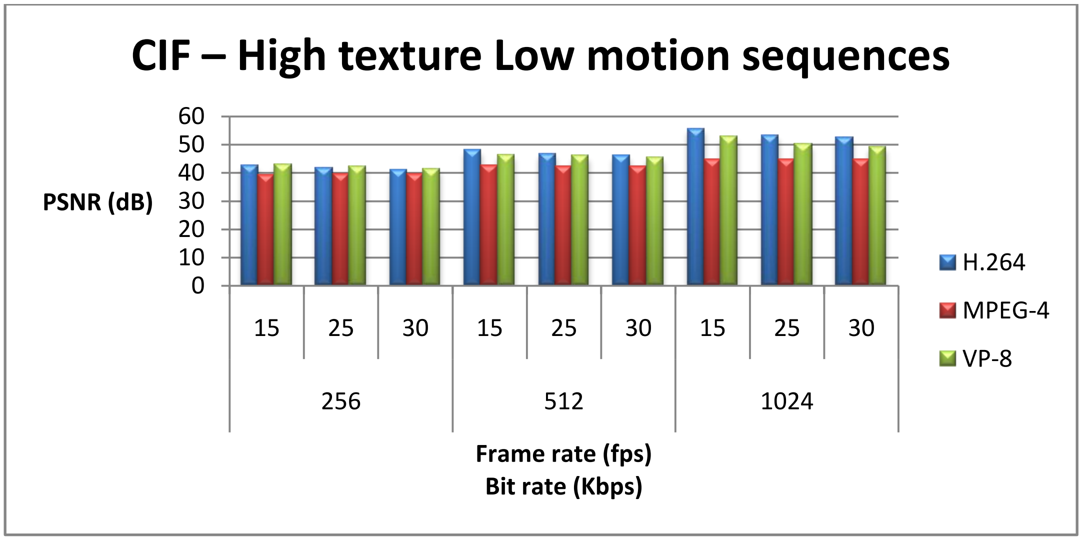

Figure 4-6 CIF – High texture Low motion sequences– PSNR

Codec	Bit Rate		
	256	512	1024
H.264	3.73	4.13	4.46
MPEG-4	3.2	3.73	3.66
VP-8	3.66	4	4.2

Table 4-7 CIF – High texture Low motion sequences – DSIS

The subjective evaluation results are shown in Table 4-7 and in Figure 4-7. The H.264 codec produced the highest satisfaction at all bit rates. VP-8 was better than MPEG-4 at all bit rates. The difference between codecs increased as the bit rate was increased.

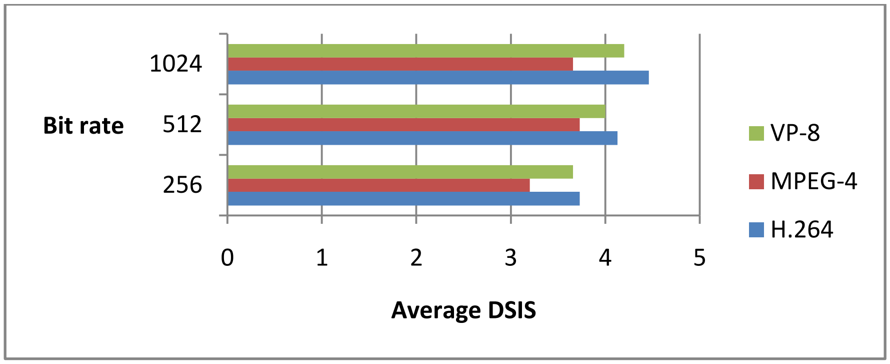

Figure 4-7 CIF – High texture Low motion sequences – DSIS

Figure 4-8, Figure 4-9 and Figure 4-10 show snapshots taken at 1024, 512 and 256 kbps respectively. As previously discussed, H.264 kept the shapes and colors close to the original. VP-8 shows better reconstruction quality than MPEG-4 where the latter caused some blur and loss of color density. As a result, H.264 achieved high reconstruction quality and got the highest subjective score.

	
ORIGINAL	H.264
	
MPEG-4	VP-8

Figure 4-8 High texture Low motion at 1024 kbps

ORIGINAL

H.264

MPEG-4

VP-8

Figure 4-9 High texture Low motion at 512 kbps

Figure 4-10 High texture Low motion at 256kbps

4.2.1.3 Low texture High motion

The average PSNR values for these video sequences are shown in Table 4-8 and in Figure 4-11. All codecs showed better PSNR by increasing the bit rate, where it decreased when increasing the frame rate. H.264 showed better PSNR than other codecs at higher bit rates. It gave approximately 1 dB better than VP-8 and 2.5 dB better than MPEG-4. Conversely, VP-8 outperformed other codecs at lower bit rates. The difference between VP-8 and other codecs at 256 kbps was less than 1 dB, while it increased to more than 1 dB better than MPEG-4 at 512 Kbps.

Bit Rate (kbps)	Frame Rate (fps)	PSNR (dB)		
		H.264	MPEG-4	VP-8
256	15	37.57	37.04	38.42
	25	37.07	36.87	37.96
	30	36.85	37.38	37.63
512	15	41.49	40.12	41.63
	25	41.17	39.45	41.35
	30	41.05	39.48	41.19
1024	15	45.56	43.76	44.76
	25	45.37	42.91	44.47
	30	45.31	42.94	44.29

Table 4-8 CIF – Low texture High motion sequences – PSNR

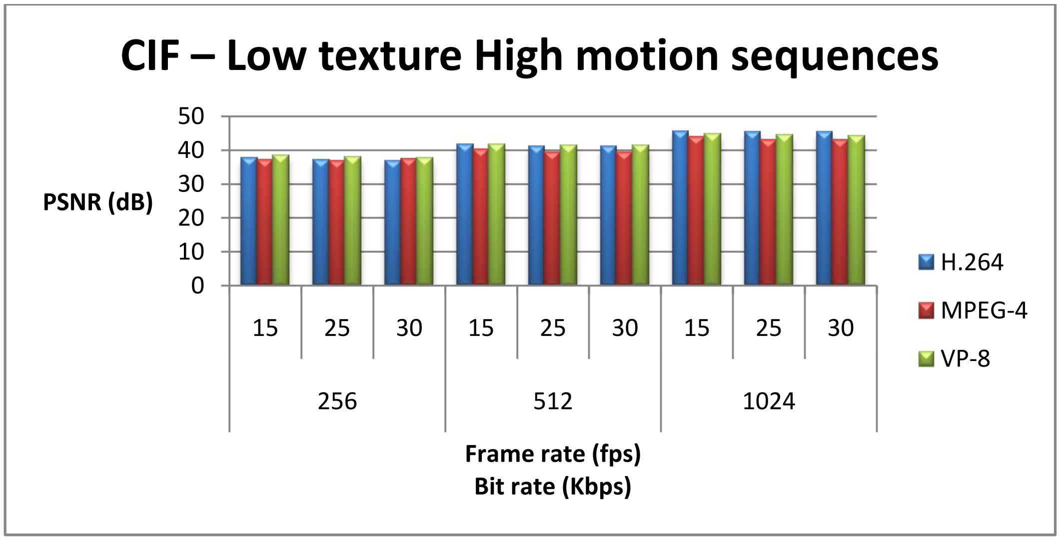

Figure 4-11 CIF – Low texture High motion sequences – PSNR

At lower bit rates all codecs gave similar PSNR results where VP-8 slightly outperformed the others. At 256 kbps, it is recommended to use VP-8 if PSNR and time are the major evaluation metrics, because it provides better PSNR with lower compression time complexity than H.264. On the contrary, subjective evaluation illustrates preference of the H.264 codec.

Codec	Bit Rate		
	256	512	1024
H.264	4.33	4.2	4.4
MPEG-4	3.93	4.13	4.06
VP-8	4.13	4.13	4.33

Table 4-9 CIF – Low texture High motion sequences – DSIS

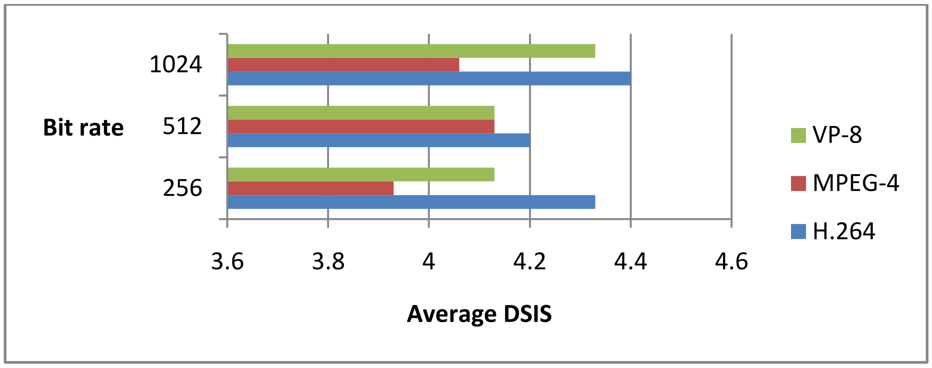

Figure 4-12 CIF – Low texture High motion sequences – DSIS

Table 4-9 and Figure 4-12 illustrate the average DSIS for this type of sequences. All codecs got high averages in the subjective evaluation results. However, it was clear that the H.264 codec got better subjective evaluation than other codecs at all bit rates. Figure 4-13, Figure 4-14 and Figure 4-15 show snapshots from the coded videos for all tested codecs. The snapshots show that all codecs have kept colors, shapes and most details, but H.264 appears to be performing better than the others.

	
ORIGINAL	H.264
	
MPEG-4	VP-8

Figure 4-13 Low texture High motion at 1024 kbps

	
ORIGINAL	H.264
	
MPEG-4	VP-8

Figure 4-14 Low texture High motion at 512 kbps

	
ORIGINAL	H.264
	
MPEG-4	VP-8

Figure 4-15 Low texture High motion at 256 kbps

4.2.1.4 Low texture Low motion

For the low texture low motion video sequences, the H.264 codec outperforms other codecs at the higher bit rates. Table 4-10 and Figure 4-16 show the average PSNR, where H.264 performed on average 2 dB better than VP-8 and was approximately 14 dB better than MPEG-4 at 1024 kbps. This gap decreased at 512 kbps, where H.264 was around 7 dB and 1 dB better compared to MPEG-4 and VP-8, respectively.

Bit Rate (kbps)	Frame Rate (fps)	PSNR (dB)		
		H.264	MPEG-4	VP-8
256	15	47.95	42.5	48.7
	25	44.36	40.44	44.51
	30	42.84	39.88	43.18
512	15	55.55	44.37	53.64
	25	50.78	44.42	49.47
	30	48.93	43.42	48.21
1024	15	62.44	44.38	60.07
	25	59.66	44.43	57.81
	30	57.42	44.44	55.36

Table 4-10 CIF – Low texture Low motion sequences – PSNR

At a lower bit rate, VP-8 performed better than other codecs. It was 4 dB better than MPEG-4 and approximately 1 dB better on average compared to H.264. VP-8 was more suitable at the lower bit rates when the PSNR evaluation metric was used. With the subjective evaluation, users preferred H.264, as shown by the score of the DSIS evaluation test where H.264 performed a little better than VP-8.

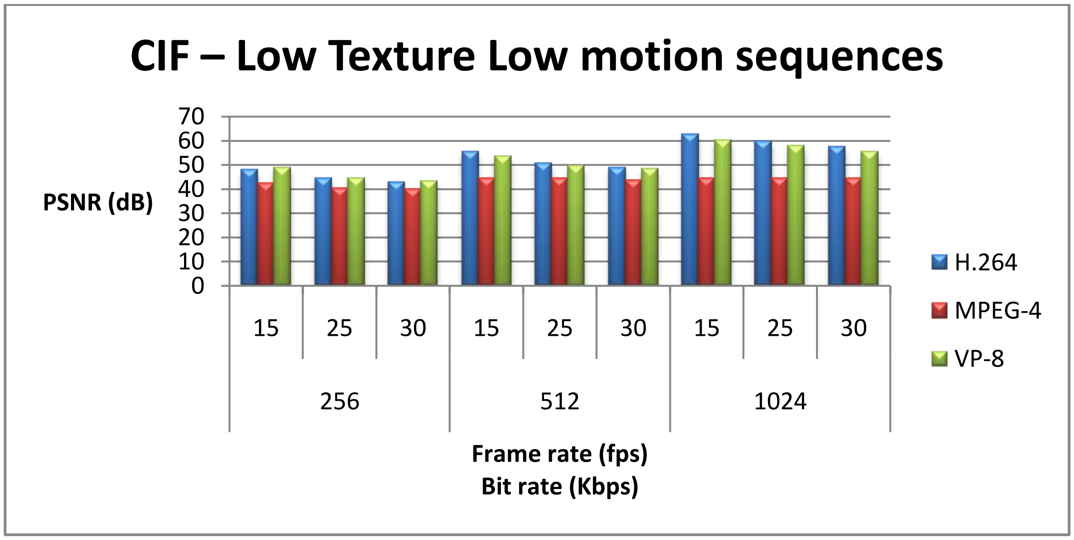

Figure 4-16 CIF – Low texture Low motion sequences – PSNR

Codec	Bit Rate		
	256	512	1024
H.264	4.2	4.33	4.66
MPEG-4	3.53	4.13	4.06
VP-8	4.13	4.26	4.6

Table 4-11 CIF – Low texture Low motion sequences – DSIS

Table 4-11 and Figure 4-17 demonstrate average DSIS. H.264 and VP-8 were almost equal with a little preference for H.264 as it was less than 1dB better than VP-8. Both codecs were better than MPEG-4 at all bit rates and frame rates.

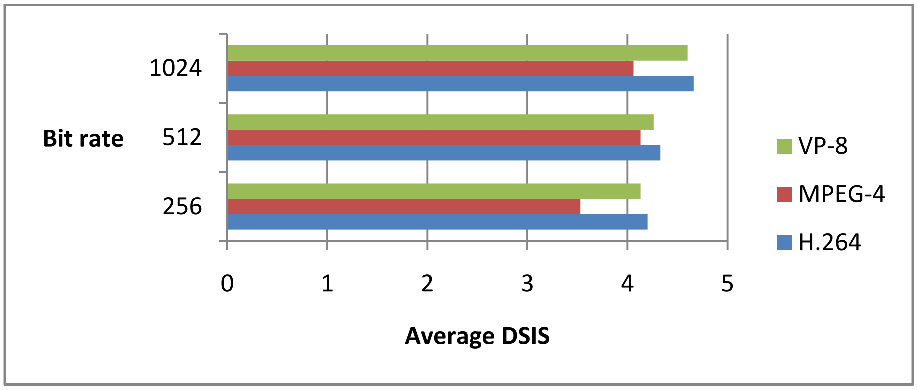

Figure 4-17 CIF – Low texture Low motion sequences – DSIS

As shown in Figure 4-18, Figure 4-19 and Figure 4-20, codecs have mostly kept the major video content details. This type of sequences is considered easy to compress with high compression performance as it does not contain too much detail or fast motion. As a result, all codecs are expected to keep the video sequence details.

	
ORIGINAL	H.264
	
MPEG-4	VP-8

Figure 4-18 Low texture Low motion at 1024 kbps

	
ORIGINAL	H.264
	
MPEG-4	VP-8

Figure 4-19 Low texture Low motion at 512 kbps

	
ORIGINAL	H.264
	
MPEG-4	VP-8

Figure 4-20 Low texture Low motion at 256 kbps

4.2.2 QCIF Format Resolution

QCIF resolution frame size is typically used for remote viewing. It is often used in on-line gaming by a player to view other team members' game information. The following subsections will examine the codecs performance for this resolution. A comparative study for different frame rates and bit rates will be presented.

4.2.2.1 High texture High motion

Bit Rate (kbps)	Frame Rate (fps)	PSNR (dB)		
		H.264	MPEG-4	VP-8
256	15	41.95	41.69	39.42
	25	41.47	42.71	39.28
	30	41.32	43.37	39.11
512	15	50.1	43.31	46.34
	25	49.5	43.5	45.85
	30	49.38	43.6	45.65
1024	15	64.89	43.32	57.47
	25	64.15	43.54	57.56
	30	63.38	43.61	57.63

Table 4-12 QCIF – High texture High motion sequences – PSNR

This type of video sequence is considered difficult for video codecs due to the content complexity and the fast scene change. The idea of compression used employs either spatial or temporal prediction. The codec needs to use the adjacent pixels for spatial prediction or previously decoded frames for temporal prediction. In this video sequences type, the codec cannot find this match easily. Therefore, it was expected that one would see a drop in codecs performance in such video sequence types.

As shown in Table 4-12 and Figure 4-21, H.264, significantly outperformed other codecs at higher bit rates. The PSNR was around 4 dB better than VP-8 and 6 dB better than MPEG-4 at 512 kbps. The difference in PSNR increased as the bit rate was increased. H.264 gave more than 20 dB and about 8 dB better fidelity than MPEG-4 and VP-8, respectively. On the other hand, MPEG-4 was slightly better than other codecs at a lower bit rate with a high frame rate, as it gave 1 dB better fidelity on average compared to H.264. The H.264 codec showed very high superiority at a higher bit rate. This performance decreased as the bit rate was increased. This appeared at 512 kbps, but it kept a significant lead. As a conclusion, although MPEG-4 performs better than H.264 at lower bit rate, the disparity is only at 256 kbps and H.264 codec is recommended for such video sequences with higher bit rates.

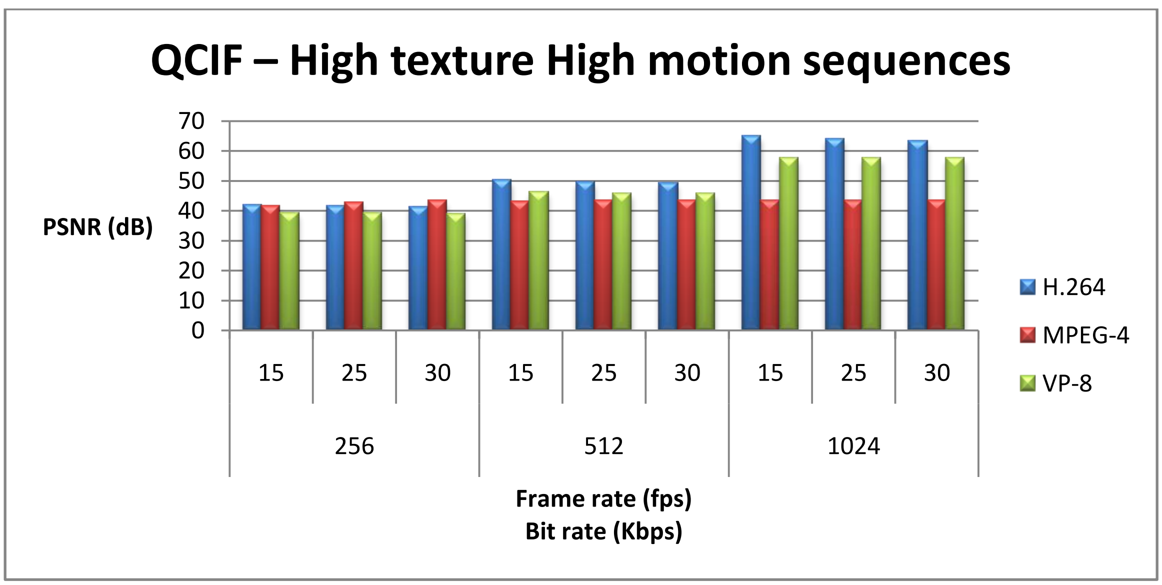

Figure 4-21 QCIF – High texture High motion sequences – PSNR

Codec	Bit Rate		
	256	512	1024
H.264	3	3.33	3.53
MPEG-4	2.4	2.4	3
VP-8	2.6	3.06	3.46

Table 4-13 QCIF – High texture High motion sequences – DSIS

Table 4-13 and Figure 4-22 demonstrate the average DSIS for this type of sequences. The results show that H.264 gave better performance than other codecs, while VP-8 performance came in second. However, results illustrate that all codecs achieved around 3 out of 5 on average in this subjective evaluation. The reason for this result is the complexity of this type of sequences and the compression difficulty that the codecs encountered.

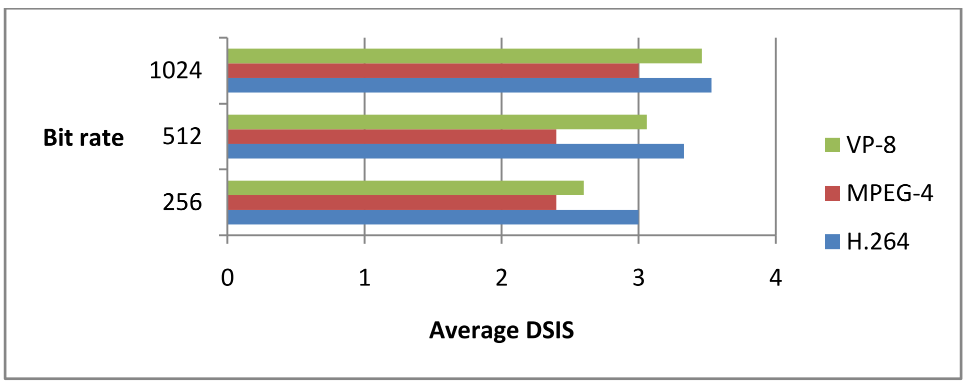

Figure 4-22 QCIF – High texture High motion sequences – DSIS

The following figures; Figure 4-23, Figure 4-24 and Figure 4-25 show snapshots of the compressed sequences. It is visually clear that all codecs produced image blurring and distortion. Text to the left-bottom of the compressed video is not clearly readable.

	
ORIGINAL	H.264
	
MPEG-4	VP-8

Figure 4-23 High texture High motion at 1024 kbps

ORIGINAL

H.264

MPEG-4

VP-8

Figure 4-24 High texture High motion at 512 kbps

	
ORIGINAL	H.264
	
MPEG-4	VP-8

Figure 4-25 High texture High motion at 256 kbps

4.2.2.2 High texture Low motion

Results for this video sequence type are shown in Table 4-14 and Figure 4-26. H.264 significantly outperformed other codecs, as it gave more than 6 dB than MPEG-4 and more than 4 dB than VP-8 at 256 kbps. The gap in performance increased as the bit rate was increased. This appeared as the difference increased to approximately 13 dB compared to MPEG-4 and 5 dB compared to VP-8 at 512 dB. At 1024 kbps, it was 19 dB on average compared to MPEG-4 and 6 dB compared to VP-8.

Bit Rate (kbps)	Frame Rate (fps)	PSNR (dB)		
		H.264	MPEG-4	VP-8
256	15	52.07	43.64	47.18
	25	49.73	43.85	45.54
	30	48.86	43.9	45.11
512	15	61.53	43.64	56.39
	25	58.27	43.85	52.18
	30	56.96	43.9	51.23
1024	15	62.83	43.65	56.58
	25	62.31	43.86	56.33
	30	62.05	43.91	56.19

Table 4-14 QCIF – High texture Low motion sequences – PSNR

The subjective evaluation results are shown in Table 4-15 and Figure 4-27. VP-8 and H.264 showed similar evaluation results at 256 kbps, where both codecs were better than MPEG-4. When increasing the bit rate, H.264 started to perform better than VP-8. Results show that the difference increased as the bit rate was increased.

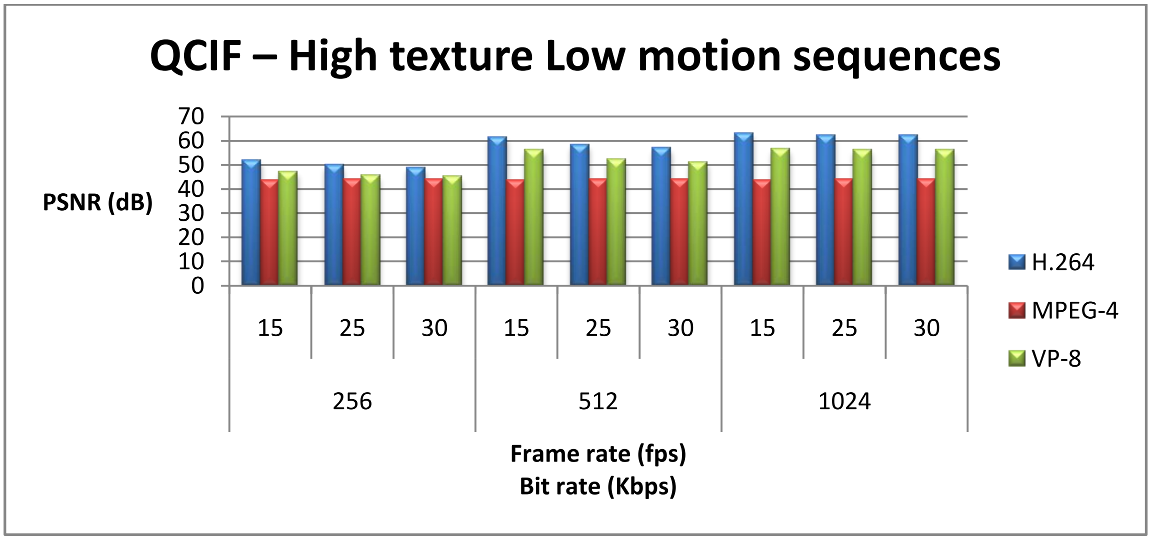

Figure 4-26 QCIF – High texture Low motion sequences – PSNR

Codec	Bit Rate		
	256	512	1024
H.264	3.13	3.26	3.66
MPEG-4	2.8	2.93	3
VP-8	3.13	3.2	3.4

Table 4-15 QCIF – High texture Low motion sequences – DSIS

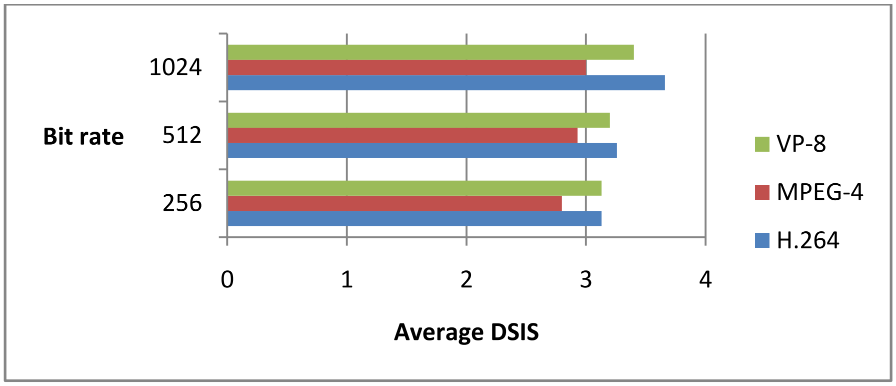

Figure 4-27 QCIF – High texture Low motion sequences – DSIS

Snapshots in Figure 4-28, Figure 4-29 and Figure 4-30 show the loss of quality in the compressed videos. It is visually clear that H.264 kept more details than other codecs. Color in H.264 is close to the original while the color difference is visible with MPEG-4 and VP-8.

	
ORIGINAL	H.264
	
MPEG-4	VP-8

Figure 4-28 High texture Low motion at 1024 kbps

	
ORIGINAL	H.264
	
MPEG-4	VP-8

Figure 4-29 High texture Low motion at 512 kbps

	
ORIGINAL	H.264
	
MPEG-4	VP-8

Figure 4-30 High texture Low motion at 256 kbps

4.2.2.3 Low texture High motion

The average PSNR values for these sequences are given in Table 4-16 and Figure 4-31. For this type of sequences, MPEG-4 showed capability close to H.264 at a lower bit rate, where the difference was less than 1 dB on average. Conversely, VP-8 gave the lowest PSNR at 256 kbps.

At higher bit rates, H.264 showed an increase in PSNR with an increase in the difference from other codecs. H.264 is about 3 dB better than MPEG-4 and more than 2 dB better than VP-8 at 512 kbps. This gap increased at 1024 kbps to be about 8 dB and 9 dB compared to VP-8 and MPEG-4, respectively.

Bit Rate (kbps)	Frame Rate (fps)	PSNR (dB)		
		H.264	MPEG-4	VP-8
256	15	42.2	41.2	39.96
	25	41.96	41.07	39.85
	30	41.9	41.58	39.65
512	15	46.96	43.81	44.47
	25	46.87	43.91	44.16
	30	46.85	43.96	43.89
1024	15	52.66	43.82	44.09
	25	52.78	43.92	43.88
	30	52.8	43.96	43.75

Table 4-16 QCIF – Low texture High motion sequences – PSNR

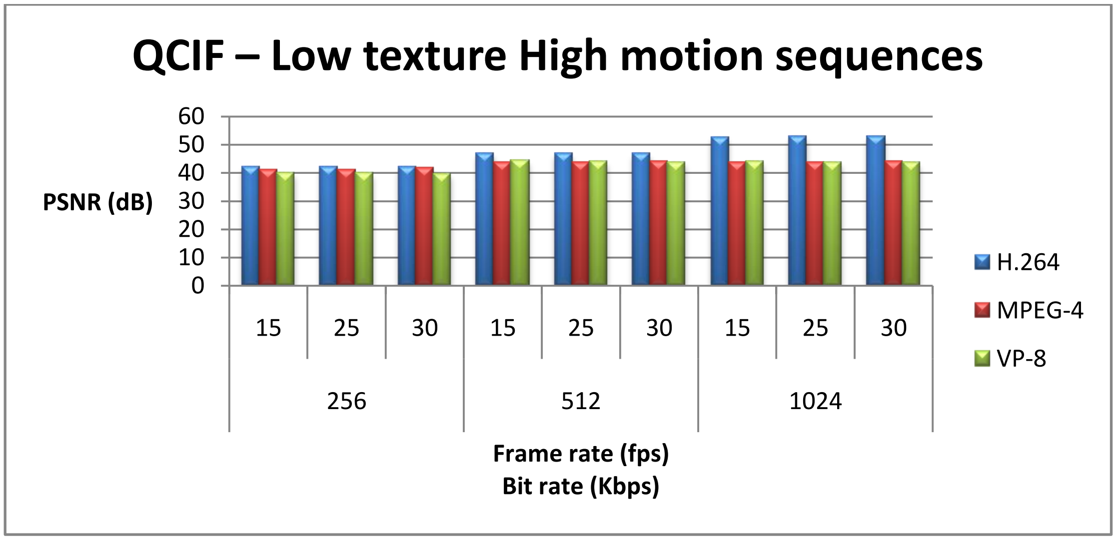

Figure 4-31 QCIF – Low texture High motion sequences – PSNR

Subjective evaluation results are illustrated in Table 4-17 and Figure 4-32. Results of H.264 and VP-8 were similar at a lower bit rate and close at higher rate. The difference increased slightly as the bit rate was increased with the H.264 preference.

Codec	Bit Rate		
	256	512	1024
H.264	3.33	3.4	3.66
MPEG-4	3.06	3.06	3.33
VP-8	3.33	3.33	3.46

Table 4-17 QCIF – Low texture High motion sequences – DSIS

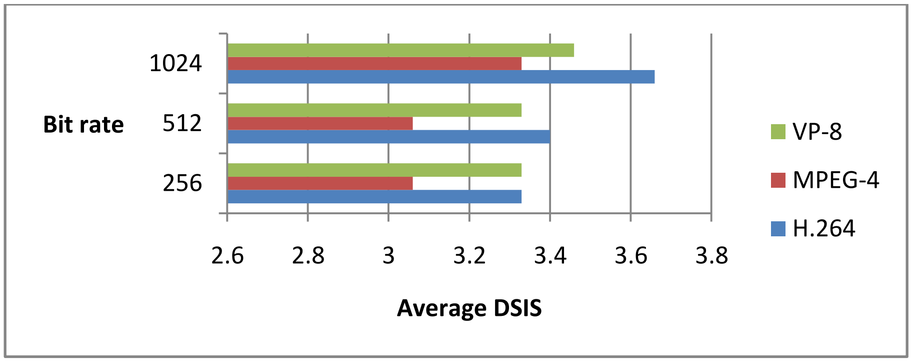

Figure 4-32 QCIF – Low texture High motion sequences – DSIS

The snapshots shown in Figure 4-33, Figure 4-34 and Figure 4-35 show the original and compressed sequences. It is visually clear that all codecs have caused loss in quality, and the text at the bottom-right is difficult to read.

	
ORIGINAL	H.264
	
MPEG-4	VP-8

Figure 4-33 Low texture High motion at 1024 kbps

ORIGINAL

H.264

MPEG-4

VP-8

Figure 4-34 Low texture High motion at 512 kbps

ORIGINAL

H.264

MPEG-4

VP-8

Figure 4-35 Low texture High motion at 256 kbps

4.2.2.4 Low texture Low motion

For this type of video sequences, H.264 showed better performance than other codecs. As seen in Table 4-18 and Figure 4-36, it was about 13 dB better than MPEG-4 and about 1 dB better, on average, compared to VP-8 at 256 kbps. However this difference clearly decreased as the frame rate was increased. At the 512 kbps bit rate, H.264 still had the superiority with the increases in the PSNR difference. It surpassed MPEG-4 and VP-8 with about 18 dB and 2 dB, respectively. This difference increased to 20 dB, on average, compared to MPEG-4 and to 4 dB compared to VP-8.

Bit Rate (kbps)	Frame Rate (fps)	PSNR (dB)		
		H.264	MPEG-4	VP-8
256	15	60.82	44.08	57.89
	25	57.32	44.15	55.73
	30	54.72	44.172	54.19
512	15	63.54	44.08	61.42
	25	62.84	44.15	60.68
	30	61.87	44.172	60.17
1024	15	64	44.08	61.78
	25	64.06	44.15	60.58
	30	63.83	44.173	58.62

Table 4-18 QCIF – Low texture Low motion sequences – PSNR

Subjective results are shown in Table 4-19 and Figure 4-37. At 256 kbps, both H.264 and VP-8 showed comparable results, both codecs were slightly better than MPEG-4. At higher bit rates, H.264 started performing better than VP-8 with a small difference at 512 kbps which is increased at 1024 kbps.

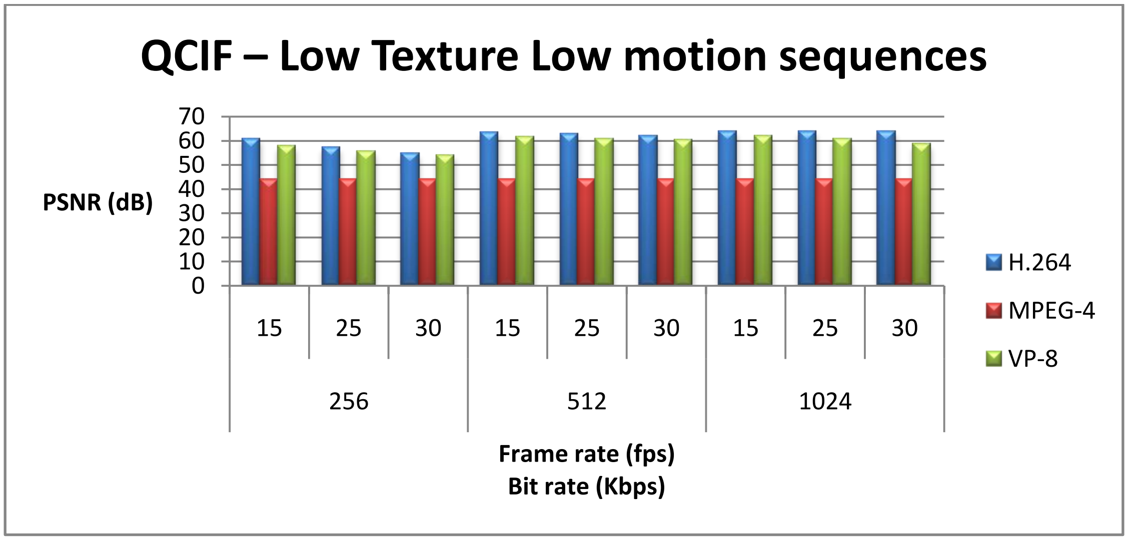

Figure 4-36 QCIF – Low texture Low motion sequences – PSNR

Codec	Bit Rate		
	256	512	1024
H.264	3.33	3.53	4
MPEG-4	3.13	2.93	3.13
VP-8	3.33	3.46	3.73

Table 4-19 QCIF – Low texture Low motion sequences – DSIS

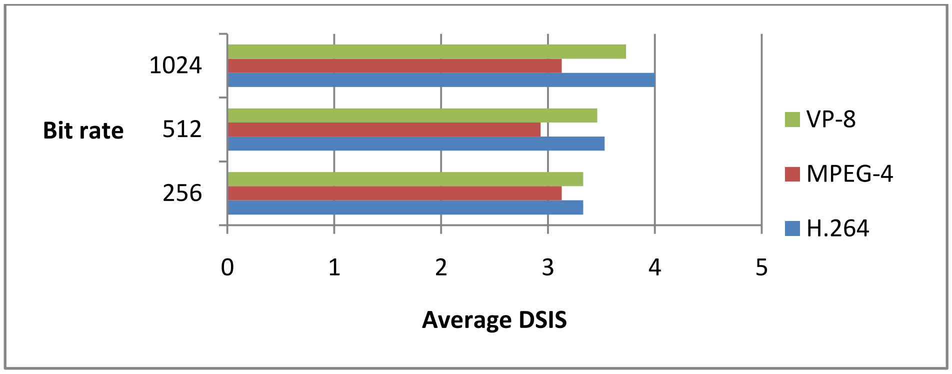

Figure 4-37 QCIF – Low texture Low motion sequences – DSIS

Figure 4-38, Figure 4-39 and Figure 4-40, show snapshots of the original and the compressed sequences. It is visually clear that all codecs caused some blurring and content loss. This type of video sequences has low content, which means that it contains small details and not much text. As a result, this blurring will not affect game streaming quality in the same way as it does in high texture sequences.

	
ORIGINAL	H.264
	
MPEG-4	VP-8

Figure 4-38 Low texture Low motion at 1024 kbps

	
ORIGINAL	H.264
	
MPEG-4	VP-8

Figure 4-39 Low texture Low motion at 512 kbps

	
ORIGINAL	H.264
	
MPEG-4	VP-8

Figure 4-40 Low texture Low motion at 256 kbps

4.3 STATISTICAL STUDY OF SUBJECTIVE TESTING RESULTS

In this chapter, the main contribution was studying and comparing the performance of video codecs for video computer graphics. Three video codecs were selected; H.264, MPEG-4 and VP-8. The comparison between these codecs was performed using subjective and objective metrics. PSNR was selected from objective metrics and DSIS from subjective metrics.

The compression performance experiments showed that H.264 often produced the highest evaluation among the codecs. Both subjective and objective metrics confirmed the superior performance of this codec. Even though H.264 showed high efficiency for both CIF and QCIF resolutions, the superiority of CIF resolution was in higher bit rates.

In addition, MPEG-4 had the best compression performance compared to other codecs at lower bit rates, CIF resolution and high texture and motion sequences. The advantage of MPEG-4 increased as frame rate was increased.

Conversely, VP-8 showed better performance over the other codecs for CIF resolution and lower bit rates with all video sequences. Nevertheless, VP-8 outperformed H.264 with an average of less than 1 dB. Although the difference was less than 1dB, this result leads to consider VP-8 as a real competitor with the H.264 codec at such compression conditions.

The PSNR and MOS results of codec comparison were shown in the preceding section. The descriptive statistical study in the section showed these simple observations: First, H.264 codec outperformed both MPEG-4 and VP-8. Second, the type of video sequence and frame rate affected MOS; the higher the motion and texture video sequences the higher the MOS.

For completeness of studying the influence of different dimensions on MOS, this section considers a wider scope of subjective video quality assessment analysis. Evaluation of perceptual visual quality under different requirements was performed including five distinctive dimensions: encoder type, video type (content), bit rate, frame rate and frame size.

The obtained subjective results were statistically analyzed, and the influence of the different dimensions on the MOS has been illustrated. The Analysis of Variance (ANOVA) is used to conduct this evaluation.

ANOVA is a statistical technical method that can be used to compare the means of two or more groups under certain assumptions (the assumptions are discussed later). There are several types of ANOVA that can be found in the literature for the purpose of comparison such as the one-way ANOVA, two-ways ANOVA, N-Way ANOVA, factorial analysis and other types, refer to [95] for details of ANOVA. To test the effect of video type (VT), Codec Type (CT), frame per second (Fps) and bit rate (BR) on MOS, three-way ANOVA was adopted in this analysis.

The analysis focuses on capturing any difference in MOS mean under the above potential factors. The model that will be tested is as follows:

$$MOS = BR + FpS + VT + \varepsilon \qquad (4-1)$$

where ε is the error, the random component.

In the above model the error is assumed normally and identically distributed with mean 0 and variance σ^2. Before applying the analysis three assumptions for ANOVA should be validated:

1- Normality assumption: we resort to large sample theory to justify the normality of our sample since the sample size is over 30.
2- Independency: the independency assumption is satisfied since each DSIS result is not affected by the others.

3- Homoscedasticity (equality of variances): the Levene's Test of Equality of Error Variances was conducted and the results are shown in Table 4-20. Results show that no major violation of this assumption is observed.

F	df1	df2	Sig.
0.902	11	36	0.548

Table 4-20 Equality of Error Variances

where Sig is the P value, it is insignificant as the p-value is greater than 0.05.

In order to perform the statistical study of the three-way ANOVA, a set of treatments are determined to study the influence of different dimensions on the MOS, the variables are Codec Type (CT), Video Type (VT) and Bit rate (BR).

The analysis results of QCIF are illustrated in Table 4-21. As shown in the table, p-values are small ($p<=0.05$), the smallness of p-values indicate that the influence of the dimensions are substantial and all have a significant effect on MOS. In addition, the magnitudes of these dimensions are indicators of the strength of the influence on the MOS. The CT impacts the MOS the most followed by VT while the least impact is that of BR.

Dimensions	Sum of Squares	Degrees of Freedom	Mean Square	F statistic	p-value
CT	1.819	2	.909	15.497	.0000
VT	1.809	3	.603	10.276	.0002
BR	.548	2	.274	4.671	.018

Table 4-21 Three-Way ANOVA on MOS of QCIF

For codecs performance evaluation, the Pairwise comparison results are shown in Table 4-22. The results show that the (p-values) are significant for CT and substantially affects the MOS between H.264 and MPEG-4, and is also highly affected between VP-8 and MPEG-4. Furthermore, positive Mean Difference value where (I is H.264 or VP-8) means that the advantage of the significance is for the codec represented by (I). On the other hand, negative value indicates that the advantage is for the codec that is represented by (J). As a result, this implies that MOS is severely affected by CT when compare H.264 or VP-8 with MPEG-4.

(I) CT	(J) CT	Mean Difference (I-J)	Std. Error	p-value	95% Confidence Interval for Difference	
					Lower Bound	Upper Bound
H264	MPEG-4	.532	.099	.000	.330	.735
	VP-8	.145	.099	.154	-.058	.348
MPEG-4	H264	-.532	.099	.000	-.735	-.330
	VP-8	-.387	.099	.001	-.590	-.185
VP-8	H264	-.145	.099	.154	-.348	.058
	MPEG-4	.387	.099	.001	.185	.590

Table 4-22 Pairwise of Video Codecs of QCIF

At the comparison of H.264 and VP-8, the experiments show that the results are insignificant (p-values>=0.05). However, Mean Difference results show that H.264 outperforms VP-8 as the positive value is when (I = H.264).

The results of the three-way ANOVA of CIF format are illustrated in Table 4-23. As shown in the table, all dimensions are significant, where p-values are almost zero. It can be noticed from the sixth column that ($p<=0.05$), this indicates that the influence of the dimensions is substantially significant for CT, VT and BR on MOS. In addition, the magnitudes of BR are less than the BR magnitude of QCIF. Therefore, this is considered as an indicator of the less strength of the influence of BR on MOS compared to the most BR influence in QCIF. In CIF format, the effect of CT, VT and BR on MOS is comparable.

Dimensions	Sum of Squares	Degrees of Freedom	Mean Square	F statistic	p-value
CT	1.520	2	.760	35.067	.000
VT	.625	3	.208	9.610	.000
BR	.805	2	.402	18.572	.000

Table 4-23 Three-Way ANOVA on MOS of CIF

The Pairwise comparison results are shown in Table 4-24. The results show that (p-values) are significant of CT and are substantially affects the MOS between H.264 and MPEG-4, MOS is also highly affected between VP-8 and MPEG-4.

Mean Difference value has either positive or negative values, where (I is H.264 and J is VP-8). If the result is positive that means the advantage of the significance is for the codec represented by (I). Otherwise, negative value indicates that the advantage is for the codec that is represented by (J). As a result, this implies that MOS is severely affected by CT when comparing H.264 or VP-8 with MPEG-4.

(I) CT	(J) CT	Mean Difference (I-J)	Std. Error	p-value	95% Confidence Interval for Difference	
					Lower Bound	Upper Bound
H264	MPEG-4	.496	.060	.000	.373	.619
	VP-8	.173	.060	.007	.050	.296
MPEG-4	H264	-.496	.060	.000	-.619	-.373
	VP-8	-.323	.060	.000	-.446	-.199
VP-8	H264	-.173	.060	.007	-.296	-.050
	MPEG-4	.323	.060	.000	.199	.446

Table 4-24 Pairwise of Video Codecs of CIF

At the comparison of H.264 and VP-8, the results show different results than in QCIF. It is clear that ($p<=0.05$) which means it is significant and MOS is severely affected by changing between the two codecs. In addition, Mean Difference results show that H.264 outperforms VP-8 as the positive value is when (I = H.264). As a conclusion, we can claim that H.264 outperformed both codecs other at CIF and QCIF resolutions.

This chapter showed that H.264 outperformed other codecs in both subjective and objective tests for most bit rates. However, the efficiency difference between H.264 and MPEG-4 was more significant than the difference between H.264 and VP-8. On the other hand, VP-8 outperformed MPEG-4 for most bit rates.

The significance of MOS was severely affected by VT, CT and BR. This shows that these dimensions (VT, CT and BR) are all important in on-line streaming and should be considered for increasing user satisfaction.

MPEG-4 was designed to support several types and contents of video sequences such as moving video (rectangular frames), video objects, animated human faces-and-bodies and static texture. MPEG-4 supports synthetic and synthetic-natural hybrid videos while H.264 was designed to support rectangular frames. However, H.264 supports small motion compensation block sizes such as 4x4 compared to a minimum size of 8x8 in MPEG-4. This built-in deblocking filter is one more advantage of H.264.

All of the above advantages of the H.264 codec led to its better results in the tests conducted in this chapter, compared to the results of MPEG-4 and VP-8. Therefore, H.264 will be the video codec of choice in the experiments conducted in subsequent chapters.

5 FAST INTER MODE SELECTION WITH REFERENCE FRAME CONTROL

5.1 INTRODUCTION

The H.264/AVC video coding standard has been chosen from Chapter 4 as the most appropriate codec for synthetic video coding. In general, H.264 is one of the most famous coding techniques. The Inter mode and Intra mode predictions are adopted in the codec. In addition, variable block sizes are available where the inter frame coding uses 16x16, 16x8, 8x16 and P8x8. The variety of block sizes increases the encoding efficiency, considering that it uses a smaller number of block sizes.

One more advantage of H.264 is adopting multiple reference frames for motion-estimation. Increasing the number of reference frames allows H.264 codec to gain more performance in video coding. Unfortunately, adopting multiple reference frames with variable block sizes makes the mechanism of choosing the best inter mode significantly increase the computational complexity.

In the literature, several techniques were presented for the estimation of reference frame motion and inter mode prediction complexity in the H.264 structure. However, these techniques did not consider wider candidates of inter modes and reference frames with significant time saving and negligible PSNR loss.

This chapter presents an efficient fast mode selection algorithm. The idea depends on grouping inter modes where each group contains the most similar modes, the group with the best mode is chosen and other modes in the group are tested. The proposed algorithm selects the best mode from the nearest reference frame. If the next nearest reference gives worse result, then there is no need to check further references.

This chapter proposes a simple and effective technique to select the best candidate inter mode and reference frame. Experimental results will be presented to show the significance of time saving in H.264 encoding.

5.2 OBSERVATIONS AND ANALYSIS

In the H.264/AVC standard, each MB in Inter-mode prediction can be divided into seven sizes called macroblock partitions. This provides higher coding efficiency by conducting motion estimation for motion compensated. The seven mode sizes mentioned above are: 16x16, 16x8, 8x16 and P8x8 with four sub-partitions. If MB is specified as P8x8, each MB can be partitioned further into 8x8, 8x4, 4x8, and 4x4 pixel blocks called sub-macroblocks, where this set of sub-macroblocks is denoted P8X8 in "Reference H.264" [42].

In the H.264/AVC standard, a macroblock mode can be chosen as follows [6]:

I-frames contain modes: Intra-4x4 and Intra-16x16.

P-frames contain modes: Intra-4x4, Intra-16x16, SKIP, Inter-16x16, Inter-16x8, Inter-8x16, Inter-8x8, Inter-8x4, Inter-4x8 and Inter-4x4 with one reference list.

B-frames contain modes: Intra-4x4, Intra-16x16, Inter-16x16, Inter-16x8, Inter-8x16, Inter-8x8, Inter-8x4, Inter-4x8 and Inter-4x4 with two reference lists.

Our proposed method is compared to the exhaustive mode decision used inH.264. The full search exhaustive method is outlined below.

1- Check whether the current MB can be considered as an intra MB, which is always the case if the MB comes from an intra frame. If the checking is positive, go to step 6; otherwise, proceed to the next step.
2- Check the SKIP mode.
3- Compute the costs and perform the motion estimation process of the modes 16x16, 16x8, and 8x16 for the current MB.

4- Select one of the four 8x8 blocks within the current MB, perform the motion estimation process and compute the costs of four smaller block-sized modes 8x8, 8x4, 4x8, and 4x4.

5- Repeat Step 4 likewise for the other three 8x8 blocks individually.

6- Perform the intra prediction procedure and compute the costs of I4MB and I16MB.

7- Among all the modes that have been checked in the previous steps, select the one that yields the minimum cost as the best mode.

H.264 outperforms the previous standard by achieving better coding efficiency. This video coding enhancement is gained by adopting several features that increase the accuracy of motion estimation. Adopting variable block sizes is one feature that provides significant gain in coding efficiency. Another adopted feature is using multiple references, which improves the video coding performance [65]. These features cause a considerable increase in computation complexity. Therefore, this chapter proposes a low-complexity macroblock decision algorithm with efficient reference frame selection.

5.3 PROPOSED METHOD

The proposed method is compared to the state of the art to show the significance of time saving compared to other proposed methods. The technique in [6] has been chosen, implemented and compared to our proposed method. The technique in [6] is mainly based on a so-called the "Normalized Motion Vector" (NMV). NMV is calculated using the temporal information and the direction indicated by the reference frame index from both lists 0 and 1.

Macroblocks are classified in five classes based on a three direction homogeneity model. Classes are: A{16x16}, B{16x16, 16x8, 8x16, P8x8}, C{16x16, 16x8}, D{16x16, 8x16} and E{16x16, 16x8, 8x16}.

The technique can be described as follows:

1- Perform motion estimation for 4x4 block size, generate NMV at 4x4 level for the current frame.
2- Calculate the vertical, horizontal and quartered "motion homogeneity" measures for each MB in the frame.
3- Determine the current MB's class based on its motion homogeneity.
4- Determine the candidate inter modes based on the selected class, the class that contains the selected mode.
5- Calculate the RDO for the candidate inter modes in the selected class.
6- Determine the best inter mode from the selected class.
7- Go to Step 3, proceed with next MB.

Our proposed method is described in the following sections.

5.3.1 Early Termination Process

As was pointed out in the preceding section, the exhaustive H.264 algorithm tests all possible modes in the case of inter frame. Larger inter block size provides larger residual amounts especially in regions with different movements. On the other hand, smaller residual amounts are normally obtained with smaller block sizes. In smooth motion, with homogenous motion and static background, there is no need to use small blocks to represent the fine level of the frame. Large block sizes are suitable and sufficient.

Sequence Name	QP	SKIP	16x16	16x8	8x16	P8x8	Intra 4x4	Intra 16x16
Grandma (QCIF)	24	45.27	17.19	7.69	7.71	20.41	1.1	0.63
	28	53.05	15.83	7.31	7.11	15.34	0.87	0.49
	32	60.22	13.51	6.42	6.48	12.15	0.76	0.46
	36	67.02	12.22	6.34	4.76	8.57	0.68	0.41
Salesman (QCIF)	24	44.01	16.17	7.69	7.82	23.37	0.6	0.34
	28	51.21	15.82	7.71	7.83	16.43	0.65	0.35

	32	59.21	15.69	5.91	7.55	10.54	0.69	0.41
	36	65.29	15.54	4.23	7.18	6.61	0.72	0.43
Foreman	24	41.8	24.79	7.36	7.31	17.19	0.81	0.74
	28	48.83	17.21	8.67	8.73	14.84	0.9	0.82
	32	56.66	16.87	8.24	7.92	8.81	0.64	0.86
	36	63.91	16.63	8.14	3.81	6.92	0.48	0.11
Coastguard	24	41.86	24.11	8.8	8.94	14.76	0.78	0.75
	28	47.95	16.29	8.56	8.69	17.24	0.74	0.53
	32	54.3	12.44	7.27	6.73	17.88	0.6	0.78
	36	57.01	12.28	7.18	3.71	19.3	0.39	0.13
News	24	45.2	16.89	7.81	7.16	21.35	1.07	0.52
	28	52.2	16.27	7.6	7.01	15.5	0.91	0.51
	32	59.58	16.17	6.34	6.57	9.84	0.71	0.79
	36	66.98	16.14	4.88	4.22	6.25	0.45	1.08
Mobile	24	39.12	12.25	7.8	7.42	32.14	0.71	0.56
	28	42.49	11.7	7.6	7.14	30.03	0.62	0.42
	32	47.66	10.28	5.8	5.17	30.25	0.51	0.33
	36	52.87	9.68	4.6	4.56	27.6	0.44	0.25
Silent	24	44.13	14.57	8.2	8.45	22.86	1.12	0.67
	28	50.33	12.09	8.02	6.32	21.87	0.78	0.59
	32	56.92	9.03	7.3	5.61	19.99	0.73	0.42
	36	64.37	7.51	5.62	4.82	16.82	0.51	0.35

Table 5-1 Mode Distribution (UNIT :%)

Many video sequences contain a degree of homogenous motion in the regions. These regions are almost always selected to be coded using large block sizes. To verify this claim, several experiments are performed for a variety of video sequences to investigate the distribution of optimal mode selected. Video sequences and mode distribution are shown in Table 5-1.

As shown in the table, it is clear that SKIP mode has the highest probability to be chosen as best mode, where a large mode such as 16x16 is chosen in high percentage. According to this study, it is clear that the SKIP mode should be checked first because its probability is high and it does not need motion cost. The mode 16x16 is also highly probable to be the best one in the encoding process. As a result, this increases the chance of an early termination to the process which saves the time of checking the remaining modes.

This study shows that there is a relationship between the best mode selected and the region homogeneity. That is, in homogeneous regions the best mode is almost always a large one and as time consuming to test all modes. On the other hand, it is difficult to decide based on a large mode and to choose it without comparing it with others.

This study suggests grouping similar modes and checking one from each group. The group that contains the best mode will be chosen as the best group. The modes in the selected group will be tested to choose the best one. An early termination process to save time is now possible, where there is no need to test other groups.

5.3.2 Reference Frame Selection

H.264/AVC adopts multiple reference frames in inter prediction. Each mode among 16x16, 16x8, 8x16 and P8x8 can choose its reference frame independently from the available number of references, but for P8x8 sub-macroblocks it should be the same one [66].

Increasing the number of reference frames normally increases the possibility of finding best motion compensated matching. Five reference frames is the largest number supported by H.264/AVC. Computational time is affected by the distance between the current MB and the best reference region for Motion Estimation. Testing five reference frames leads to high complexity motion estimation and increased encoding time.

Sequence Name	Reference 0	Reference 1	Reference 2	Reference 3	Reference 4
Grandma (QCIF)	78.9	9.6	6.2	3.1	2.2
Salesman (QCIF)	81.3	6.8	5.7	4.6	1.6
Foreman	82.9	12.3	2.1	1.7	1
Coastguard	76.5	15.3	5.7	1.4	1.1
News	81.2	8.4	4.2	3.5	2.7
Mobile	62.5	17.8	9.4	7.1	3.2
Silent	69.7	15.7	9.2	4.1	1.3
Average	76.14	12.27	6.07	3.64	1.87

Table 5-2 Reference Frame Selection Probabilities (UNIT :%)

The first reference frame is the most likely candidate to be the most identical frame to the current one. Searching other frames normally gives higher Lagrangian values. To check this assumption, intensive experiments were conducted to test the best reference frame. Different sequence types (CIF and QCIF) with different homogeneities and motion speeds were tested. As shown in Table 5-2, testing all possible references consumes a lot of time and the probability of finding a better reference match is decreased by testing earlier references.

By using the observations in the discussion above, it is possible to save time by terminating the process of testing earlier when the matching result is going worse. The proposed algorithm will not ignore the possibility of finding a mode in other reference frames that may give better results, but it reduces the search space.

5.3.3 Overall Procedure

In the preceding subsections, the ideas of grouping inter modes and checking the most probable reference frames were shown. In the proposed algorithm, there is no need to check all modes for all reference frames. It groups all similar modes into subgroups such as intra, large inter, small inter and SKIP modes, where it is sufficient to check one mode from each group to find the best mode.

After the mode is selected for the first reference frame, and depending on experimental results such as those in Table 5-2, the closest reference frame normally provides the best match. If the next reference frame has fewer matches, then the farther frames will have even fewer matches, and it is possible to skip checking them. For completeness, the proposed procedure of the algorithm is described in full as follows:

0- Groups are G1{Intra modes}, G2{Inter 16x16, 16x8, 8x16}, G3{Inter 8x8, 8x4, 4x8, 4x4}, G4{SKIP mode}. k=0.
1- If the current MB is an Intra MB? , go to Step 7.
2- Compute the cost of SKIP mode, and compute the cost for first mode in G2 and G3 with first reference frame (k).
3- If best mode from Step (2) is SKIP mode, SKIP mode is the best mode, go to Step (7). Otherwise, choose the group that contains the mode of minimum cost.
4- Test the second mode in the selected group. If it is better, continue with the remaining modes in the group and find the best. Otherwise, go to Step (5).
5- Compute the cost of the best mode in reference frame (k+1).
6- If cost in frame (k+1) is better, continue with the remaining reference frames. Otherwise, go to Step (7).
7- Check Intra modes.

The above algorithm has an advantage that should be noted. It is possible to combine it with any fast intra algorithm to achieve higher speed up.

To verify the effectiveness of our proposed algorithm, we compared it with exhaustive search and an algorithm proposed by Liu, Shen, and Zhang [6].

5.4 EXPERIMENTAL RESULTS

Experimental results of the proposed algorithm show significant complexity reduction with negligible loss of coding efficiency. To verify the enhanced performance of the proposed algorithm fairly, the proposed algorithm was evaluated on various video sequences.

Parameter Name	Value
Number of Reference frames	5
Encoding Frame Number	150
Motion Vector Search Range	16
Motion Vector Resolution	¼
RDO/CABAC	On
Group of Pictures	IPPP and IBBP

Table 5-3 Simulation Parameters

The video sequences contained different motions and frame types, two sequences with QCIF frames including Salesman, Grandma and six sequences of CIF types including News, Silent, Paris, Foreman, Coastguard, and Mobile. The modes were tested with (QP=24, 28, 32, 36). These parameters and the tested video sequences have been chosen according to the parameters and sequences used in the benchmark paper [6] for comparison fairness. All tests in the experiments were run on an Intel Centrino Duo 2.0 GHz with 2 GB RAM, and the OS used was Windows XP.

To verify the proposed algorithm, H.264/AVC JM 9.4 was used. The fast inter mode algorithm proposed by Liu, Shen, and Zhang [6] was implemented with the same parameters as our algorithm to gain a fair comparison. All simulation parameters are presented in Table 5-3.

The metrics used to evaluate the achieved performance are: (ΔPSNR) which refers to the Peak Signal to Noise Ratio difference, (ΔBitrate %) which refers to the percentage of bit rate difference (increased) and (ΔTime %) for the percentage of time decrease in (1-5). All metrics are used to evaluate the differences between the proposed algorithm and the algorithm in [6] with the reference encoder. The metric of time percentage is defined as:

$$\Delta\mathrm{Time}(\%) = \frac{\mathrm{Time}_{\mathrm{Ref.}} - \mathrm{Time}_{\mathrm{Pro.}}}{\mathrm{Time}_{\mathrm{Ref.}}} \times 100 \qquad (1-5)$$

Where: $\mathrm{Time}_{\mathrm{Ref.}}$ is the coding time used for reference software.

$\mathrm{Time}_{\mathrm{Pro.}}$ is the coding time used for the proposed or [6] algorithms.

To evaluate the performance of the proposed algorithm, the averages of PSNR and time of the four QP values are calculated for the three methods: the proposed algorithm, [6] algorithm and the reference encoder. The gained values are shown in Table 5-4 for IPPP and in Table 5-5 for IBBP.

The proposed algorithm achieved better speedup with negligible loss of image fidelity and a minimal increase in average bit rate. It is clear from Table 5-4 that the proposed algorithm decreased the total encoding time by 59.88% on average for IPPP, while the loss of PSNR was 0.09 and bit rate increase was 2.79 on average. The proposed algorithm achieved large reductions of encoding time for different sequences with 61% as the highest, obtained for Foreman, and least was 53.29%, obtained for Grandma, while [6] shows more degradation for sequences of CIF types such as Mobile and Foreman.

IPPP						
Sequence	Proposed Algorithm			Algorithm in [6]		
	ΔBit rate (%)	ΔPSNR (dB)	ΔTS(%)	ΔBit rate (%)	ΔPSNR (dB)	ΔTS(%)
Salesman (QCIF)	-3.79	-0.12	59.43	11.5	-0.08	43.6
Grandma (QCIF)	-4.75	-0.06	53.29	9.5	-0.018	45.7
News (CIF)	1.24	-0.07	60.55	5.9	-0.114	49.2
Silent (CIF)	7.63	-0.03	60.09	6.09	-0.082	49.6
Paris (CIF)	4.21	-0.1	59.41	3.4	-0.072	41
Foreman (CIF)	15.2	-0.14	61.04	14.9	-0.049	37.7
Coastguard (CIF)	-3.23	-0.1	59.24	0.5	-0.022	33.1
Mobile (CIF)	5.81	-0.11	65.98	1.18	-0.073	28.2
Average	2.79	-0.09	59.88	6.62	-0.06	41.01

Table 5-4 Performance Comparison for the Proposed Algorithm (IPPP)

Table 5-5 shows that the proposed algorithm can reduce the IBBP encoding time by 60.24% on average compared to 39.71% for [6].The efficiency loss in terms of PSNR is 0.08 dB with a 1.8%increment in bit rate. More encoding time is saved in IBBP with negligible increment of bit rate and negligible loss of PSNR.

It is clear from Table 5-4 and Table 5-5 that our proposed algorithm achieves higher savings of encoding time by saving at least 53.29% for IPPP and 54.83% for IBBP compared to 28.2% for IPPP and 28.8% for IBBP obtained by algorithm in [6]. The higher gain of time reduction with the small loss of coding efficiency makes the proposed algorithm more suitable for fast inter mode coding especially for applications that consider total encoding time.

IBBP						
Sequence	Proposed Algorithm			Algorithm in [6]		
	ΔBit rate (%)	ΔPSNR (dB)	ΔTS(%)	ΔBit rate (%)	ΔPSNR (dB)	ΔTS(%)
Salesman (QCIF)	3.59	-0.13	59.74	4.7	-0.03	48.9
Grandma (QCIF)	-2.99	-0.01	61.92	0.3	-0.03	51.6
News (CIF)	-1.13	-0.1	57.73	2.7	-0.126	41.6
Silent (CIF)	-2.73	-0.04	64.12	0.42	-0.034	43.4
Paris (CIF)	-1.45	-0.11	61.99	11.3	-0.13	39.1
Foreman (CIF)	-0.76	-0.09	66.34	0.93	-0.068	33.6
Coastguard (CIF)	6.04	-0.16	55.22	0.53	-0.063	30.7
Mobile (CIF)	13.85	-0.01	54.83	2.4	-0.078	28.8
Average	1.80	-0.08	60.24	2.91	-0.07	39.71

Table 5-5 Performance Comparison for the Proposed Algorithm (IBBP)

6 TRADEOFF BETWEEN FRAME-RATE AND RESOLUTION

6.1 INTRODUCTION

A tradeoff between the frame rate and resolution is recommended for each bit rate. The tradeoff is expected to offer good and clear motion of the game components during playing and provide appropriate resolution.

As discussed in Chapter 4, many experiments have been conducted to evaluate the performance of video compression techniques. That chapter concluded that the H.264 codec is mostly the best suited codec for synthetic video.

Most studies in the literature for codecs and compression assessment have focused on fixed spatial and temporal resolution, while limited research focused on assessment of different spatial and temporal resolutions.

This chapter reports on a wider range of video resolutions and motion speeds. The visual quality of the compressed video sequences was perceptually evaluated using the DSIS assessment metric. Extensive subjective viewing tests to evaluate the resolution were performed.

This chapter performs a comparative study to give recommendations for the most suitable and recommended tradeoff between frame rate, frame resolution and bit rate. Four ten-second video sequences are used in the experiments. All video sequences were captured from the "World of Warcraft" game. The video sequences have different spatial and temporal contents, which vary from low texture content with low object motion to high texture with high motion.

All video sequences were compressed using an FFmpeg H.264 encoder at bit rates of 256, 512 and 1024 kbps. The frame sizes are QCIF, CIF and 4CIF with frame rates of 15, 20, 25 and 30 frames per second. Each video sequence has 36 different coding scenarios over each bit rate, frame rate and frame size.

For the subjective evaluation, all video sequences were displayed at the same resolution and frame rate. The frame size 4CIF and frame rate 30 fps were chosen for all tests. The lower resolutions were upsampled to 4CIF and lower frame rates were repeated to become 30 fps. The process of upsampling and frame repetition was conducted using the H.264/AVC 6-tap half-sample interpolation filter and frame repeat [19].

All subjective test experiments were conducted using the method of double stimulus impairment scale variant II. This method is recommended in ITU-R Recommendation BT-500-11 [17]. The full details of DSIS and subjective assessment methods were discussed in Section2.1.

6.2 SUBJECTIVE VIEWING TEST

Considering the four different video sequences with the combinations of bit rates, frame rates and frame sizes, there is a total of 144 different scenarios. Each scenario is tested and evaluated using DSIS assessment measurement. The mean opinion scores (MOS) are then calculated for 15 viewers. All viewers were selected to be university students who have good knowledge of English. Additionally, the scaling and scoring mechanism was explained to the viewers to ensure vote reliability. Results in detail will be described and discussed in the following sections.

A variety of motion speeds and textural contents was considered in the capturing process. DSIS with five levels of quality scales was used. The method displays the original sequence followed by the compressed. This process is repeated once before the viewers can score the perceptual quality. The viewers' perceptual quality is scaled in five levels describing the video quality score. The scores are 1, 2, 3, 4 and 5, which represent 'bad', 'poor', 'fair', 'good' and 'excellent', respectively. All tests in the experiments were run on an Intel Centrino Duo 2.0 GHz with 2 GB RAM, and the OS used was Windows XP. Table 6-1 shows the compression parameters used in the study.

Parameter Name	Value
Aspect Ratio	4:3
Group of Pictures	IPPP
QP	28
RDO	Yes
CABAC	No
B Slice	No
Sample Depth	8 bits

Table 6-1 Compression Parameters

6.2.1 4CIF Format Resolution

6.2.1.1 High Texture High Motion

Bit Rate (kbps)	Frame Rate (fps)	MOS
256	**15**	2
	20	1.66
	25	1.33
	30	1.33
512	**15**	3
	20	3.33
	25	3
	30	2.33
1024	**15**	4.60
	20	4.33
	25	4.66
	30	4.66

Table 6-2 4CIF – High texture High motion sequences – H.264/DSIS

The subjective evaluation results are shown in Table 6-2 and Figure 6-1. Results show a decrease in MOS as the frame rate was increased at the lower bit rate. The viewers seemed to be more satisfied by the quality when the frame rate was set to 15 fps at the 256 kbps bit rate. Increasing frame rate at a lower bit rate degraded the quality.

Conversely, at higher bit rates, MOS was more comparable at the different frame rates. At 512 kbps, MOS values were similar at lower frame rates greater than 30 fps. These results led to regarding lower frame rates as better quality at 512 kbps.

At 1024 kbps the MOS was mostly similar for all frame rates. The frame rate did not affect the quality. This result may show the way to save the bandwidth at the limited network capability by decreasing frame rate as it has not affected the quality based on this subjective evaluation.

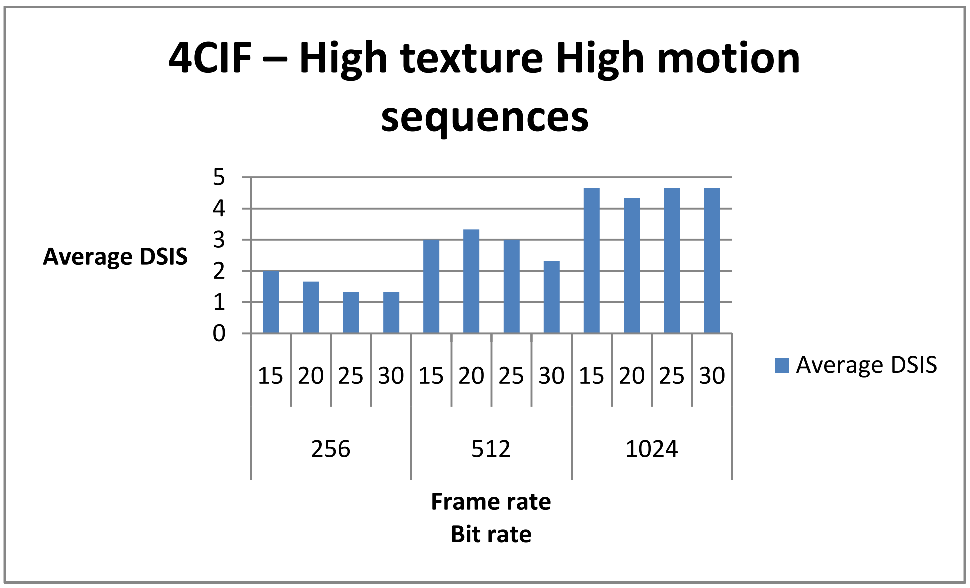

Figure 6-1 4CIF – High texture High motion sequences – H.264/DSIS

Figure 6-2, Figure 6-3, Figure 6-4 and Figure 6-5 show snapshots of the compressed and original sequences. MOS and snapshots show that higher bit rates gave the best MOS regardless of the frame rate. Lower bitrates caused slightly more blurring than the higher. Overall, all snapshots show slight quality degradation.

Original

256 kbps	512 kbps	1024 kbps
		

Figure 6-2 4CIF - High texture High motion - H.264/15 fps

Original

		
256 kbps	512 kbps	1024 kbps

Figure 6-3 4CIF - High texture High motion - H.264/20 fps

Original

		
256 kbps	512 kbps	1024 kbps

Figure 6-4 4CIF - High texture High motion - H.264/25 fps

Original

		
256 kbps	512 kbps	1024 kbps

Figure 6-5 4CIF - High texture High motion - H.264/30 fps

6.2.1.2 High Texture Low Motion

Results for this category are illustrated in Table 6-3 and

Figure 6-6. Results show that the MOS increased as the bit rate was increased. MOS, on average, was 2, 3.75, and 4.66 at bit rates 256, 512 and 1024, respectively. With respect to decreases, MOS decreased as frame rate was increased at the same bit rate. For example, at 256 kbps, MOS degraded from 2.66 at 15 fps to 1.33 at 30 fps.

At higher bit rates an increase in frame rate also resulted in a decrease in MOS. However, the quality degradation was relatively smaller. In addition, the gap between MOS for frame rates decreased as the bit rate was increased. As a result, it is possible to save bandwidth and get better visual quality by decreasing frame rate while transmitting video sequences with high texture and low motion.

Bit Rate (kbps)	Frame Rate (fps)	MOS
256	15	2.66
	20	2.33
	25	1.66
	30	1.33
512	15	4.33
	20	3.66
	25	3.33
	30	3.66
1024	15	4.93
	20	4.66
	25	4.66
	30	4.33

Table 6-3 4CIF – High texture Low motion sequences – H.264/DSIS

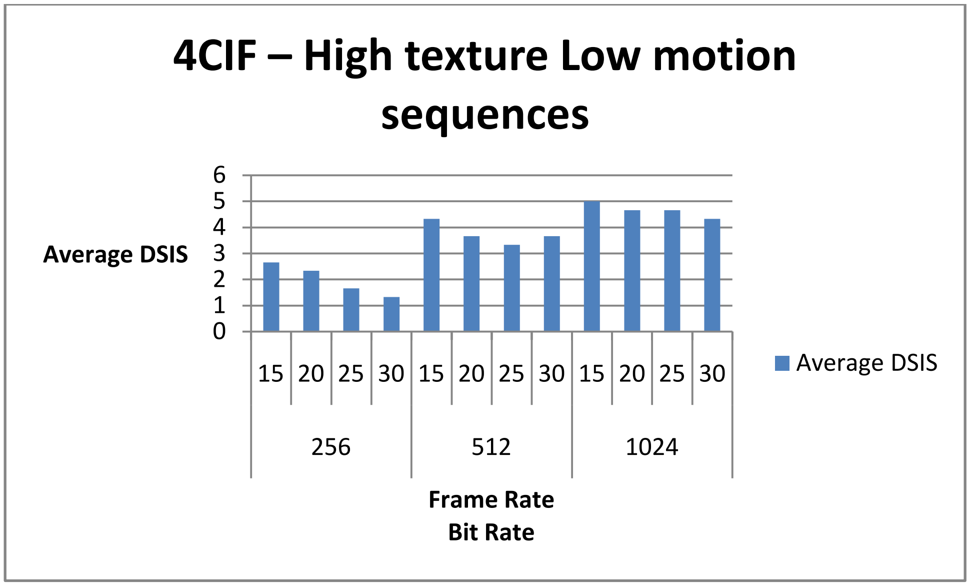

Figure 6-6 4CIF – High texture Low motion sequences – H.264/DSIS

The snapshots in Figure 6-7, Figure 6-8, Figure 6-9 and Figure 6-10 illustrate the differences in compressed video quality at the different bit rates.

Original

		
256 kbps	512 kbps	1024 kbps

Figure 6-7 4CIF - High texture Low motion - H.264/15 fps

Original

		
256 kbps	512 kbps	1024 kbps

Figure 6-8 4CIF - High texture Low motion - H.264/20 fps

<table>
<tr><td colspan="3">
</td></tr>
<tr><td colspan="3">Original</td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td>256 kbps</td><td>512 kbps</td><td>1024 kbps</td></tr>
</table>

Figure 6-9 4CIF - High texture Low motion - H.264/25 fps

Original

		
256 kbps	512 kbps	1024 kbps

Figure 6-10 4CIF - High texture Low motion - H.264/30 fps

6.2.1.3 Low Texture High Motion

As illustrated in Table 6-4 and Figure 6-11, results show increases in MOS as the bit rate was increased. H.264 showed higher MOS results which reached 4.66 at 1024 kbps bit rate.

The results were smaller at lower bit rates as it was, on average, 3.75 and 1.8 at bit rates 512 and 256, respectively. With respect to decreases, MOS decreased as frame rate increased at the same bit rate. At 256 kbps, MOS decreased from 2.66 at 15 fps to 1.33 at 30fps.

Bit Rate (kbps)	Frame Rate (fps)	MOS
256	15	2.66
	20	1.66
	25	1.66
	30	1.33
512	15	4
	20	4
	25	3.33
	30	3.66
1024	15	4.66
	20	4.66
	25	4.53
	30	4.46

Table 6-4 4CIF – Low texture High motion sequences – H.264/DSIS

Also the frame rate and MOS results showed an inverse relationship at higher bit rate. However, the degradation of quality was higher at lower bit rates, and the gap of quality degradation decreased as bit rate was increased. As a result, it is recommended to save bandwidth and get better visual quality by using lower frame rates when using video sequences of this type.

Captured snapshots in Figure 6-12, Figure 6-13, Figure 6-14 and Figure 6-15 show the differences between the compressed video sequences at different bit rates and frame rates. The quality was affected more at lower bit rates, where it was less distorted at higher bit rates.

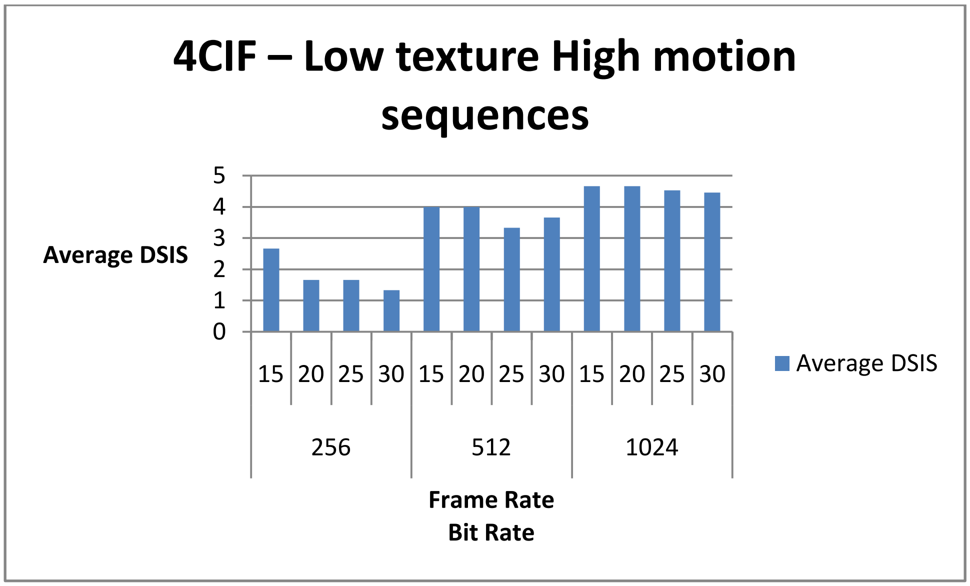

Figure 6-11 4CIF – Low texture High motion sequences – H.264/DSIS

<table>
<tr><td colspan="3">
</td></tr>
<tr><td colspan="3">Original</td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td>256 kbps</td><td>512 kbps</td><td>1024 kbps</td></tr>
</table>

Figure 6-12 4CIF - Low texture High motion - H.264/15 fps

Original

		
256 kbps	512 kbps	1024 kbps

Figure 6-13 4CIF - Low texture High motion - H.264/20 fps

Original

		
256 kbps	512 kbps	1024 kbps

Figure 6-14 4CIF - Low texture High motion - H.264/25 fps

Original

256 kbps

512 kbps

1024 kbps

Figure 6-15 4CIF - Low texture High motion - H.264/30 fps

6.2.1.4 Low Texture Low Motion

As shown in Table 6-5 and Figure 6-16, MOS increased as the bit rate was increased regardless of the change in frame rate. At lower bit rates, the best MOS was at 15 fps, which was the lowest tested frame rate. The result decreased at 20 fps by almost half. However, MOS was 2.66 at 25 fps, it was better at 20 fps and best at 15 fps. The worst MOS result was at 30 fps where it was only 1.33.

At 512 kbps, MOS produced the worst result of 2.33 at 25 kbps compared to 3 at 15 and 20 kbps. The highest MOS was at 30 fps with MOS equal to 3.33. On the other hand, at 1024 kbps, MOS result decreased as the frame rate was increased. The result was superior at 15 fps with 4.66. MOS degraded at higher frame rates and it was 3.8 at 30 fps.

Bit Rate (kbps)	Frame Rate (fps)	MOS
256	15	3
	20	1.66
	25	2.66
	30	1.33
512	15	3
	20	3
	25	2.33
	30	3.33
1024	15	4.66
	20	4
	25	3.93
	30	3.8

Table 6-5 4CIF – Low texture Low motion sequences – H.264/DSIS

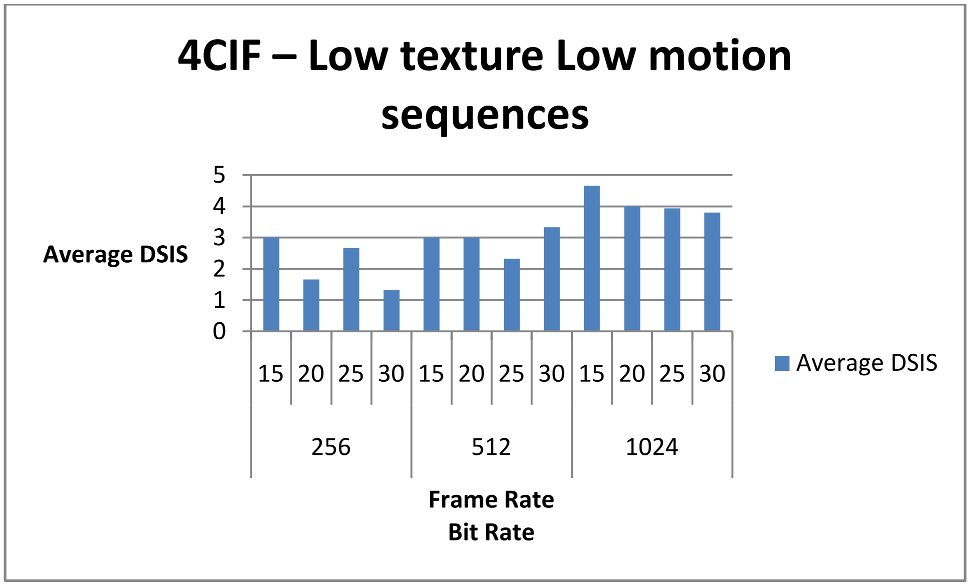

Figure 6-16 4CIF – Low texture Low motion sequences – H.264/DSIS

The snapshots shown in Figure 6-17, Figure 6-18, Figure 6-19 and Figure 6-20 are the compressed video sequences compared to the original. All sequences show acceptable quality with a few differences.

Original

		
256 kbps	512 kbps	1024 kbps

Figure 6-17 4CIF - Low texture Low motion - H.264/15 fps

Original

		
256 kbps	512 kbps	1024 kbps

Figure 6-18 4CIF - Low texture Low motion - H.264/20 fps

Original

		
256 kbps	512 kbps	1024 kbps

Figure 6-19 4CIF - Low texture Low motion - H.264/25 fps

		
Original		
		
256 kbps	512 kbps	1024 kbps

Figure 6-20 4CIF - Low texture Low motion - H.264/30 fps

6.2.2 CIF Format Resolution

6.2.2.1 High Texture High Motion

As illustrated in Table 6-6 and Figure 6-21, MOS was the lowest at 256 kbps where it was better at higher bit rates. All frame rates with 256 kbps bit rate were similar except for the 30 fps rate where MOS produced a worse result. The quality at this bit rate decreased from 1.66 to 1.33.

Bit Rate (kbps)	Frame Rate (fps)	MOS
256	15	1.66
	20	1.66
	25	1.66
	30	1.33
512	15	2.33
	20	3.33
	25	2.66
	30	2.66
1024	15	2.66
	20	2.66
	25	3
	30	3.66

Table 6-6 CIF – High texture High motion sequences – H.264/DSIS

At higher bit rates, 512 kbps had the highest MOS at 20 fps. In spite of the frame rate of 20, MOS increased as the frame rate was increased. However, the 20 fps was the highest quality result at 512 kbps, so it is recommended for this type of video sequences.

At 1024 kbps and despite of the resemblance of MOS at 15 and 20 fps, it is clear that the quality increased each time the frame rate was increased. MOS result was the highest at 30 fps with 3.66 compared to 2.66 at 15 fps.

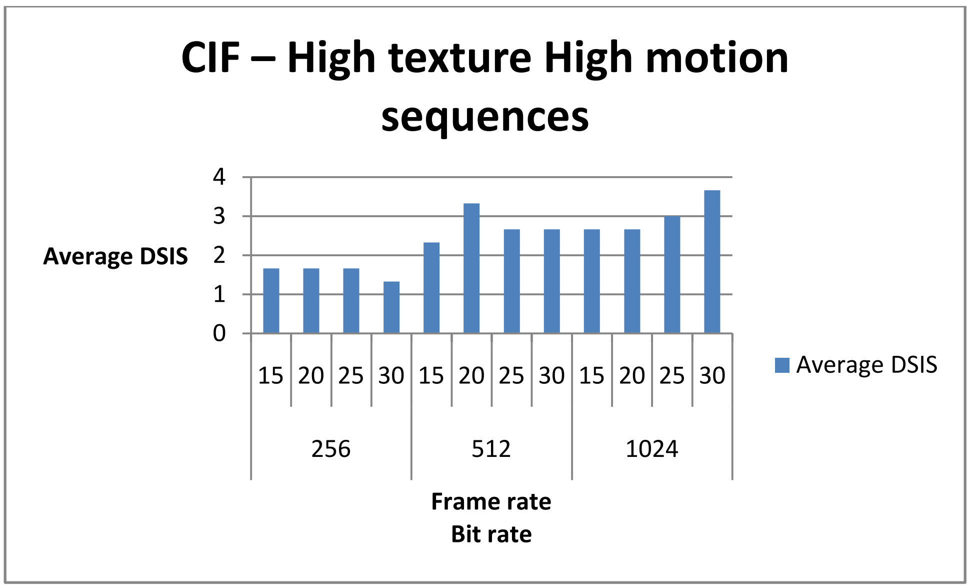

Figure 6-21 CIF – High texture High motion sequences – H.264/DSIS

The snapshots of the sequences are shown in Figure 6-22, Figure 6-23, Figure 6-24 and Figure 6-25 which present these snapshots of the different bit rates and frame rates.

		
Original		
		
256 kbps	512 kbps	1024 kbps

Figure 6-22 CIF - High texture High motion - H.264/15 fps

<table>
<tr><td colspan="3">
</td></tr>
<tr><td colspan="3">Original</td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td>256 kbps</td><td>512 kbps</td><td>1024 kbps</td></tr>
</table>

Figure 6-23 CIF - High texture High motion - H.264/20 fps

Original

		
256 kbps	512 kbps	1024 kbps

Figure 6-24 CIF - High texture High motion - H.264/25 fps

<table>
<tr><td colspan="3">
</td></tr>
<tr><td colspan="3">Original</td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td>256 kbps</td><td>512 kbps</td><td>1024 kbps</td></tr>
</table>

Figure 6-25 CIF - High texture High motion - H.264/30 fps

6.2.2.2 High Texture Low Motion

Results of this video sequence type are illustrated in Table 6-7 and Figure 6-26. MOS increased slightly as the bit rate was increased. At 256 kbps, the MOS result was, on average, 2.5 where it was around 3 and 3.16 at 512 kbps 1024 kbps, respectively.

At a lower bit rate, the quality was best at 20 and 30 fps where it decreased at 15 and 25 fps. At 512 kbps, MOS was the least at 20 fps with 2.66. Both 15 and 25 fps produced a similar MOS of 3 and the best quality result was at 30 fps with 3.33.

Bit Rate (kbps)	Frame Rate (fps)	MOS
256	15	2.33
	20	2.66
	25	2.33
	30	2.66
512	15	3
	20	2.66
	25	3
	30	3.33
1024	15	3
	20	3.33
	25	3
	30	3.33

Table 6-7 CIF – High texture Low motion sequences – H.264/DSIS

At the higher bit rate of 1024 kbps, 15 and 25 produced the same MOS with 3 compared to 3.33 at 20 and 30 fps. It is noted that the frame rates 15 and 25 at both bit rates 512 and 1024 produced the same quality evaluation. As a result, increasing the bit rate from 512 to 1024 had no effect on the user judgment and evaluation.

The same situation occurred with 30 fps at both 512 and 1024 kbps, where the MOS result was 3.33 with no effect of doubling the bit rate. The only case where MOS increased between the two bit rates was at 20 fps. This may lead to concluding that increasing the bit rate is not necessary regardless of using low or high frame rates.

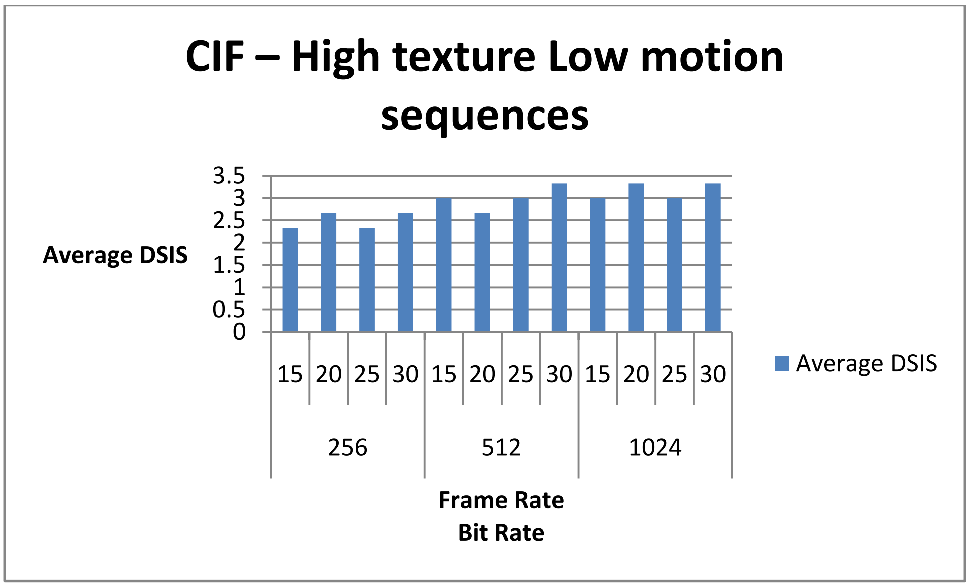

Figure 6-26 CIF – High texture Low motion sequences – H.264/DSIS

To complete the comparison, the snapshots of the compressed video sequences are shown in Figure 6-27, Figure 6-28, Figure 6-29 and Figure 6-30 with the original captured video.

<table>
<tr><td colspan="3">
</td></tr>
<tr><td colspan="3">Original</td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td>256 kbps</td><td>512 kbps</td><td>1024 kbps</td></tr>
</table>

Figure 6-27 CIF - High texture Low motion - H.264/15 fps

Original

		
256 kbps	512 kbps	1024 kbps

Figure 6-28 CIF - High texture Low motion - H.264/20 fps

<table>
<tr><td colspan="3">
</td></tr>
<tr><td colspan="3">Original</td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td>256 kbps</td><td>512 kbps</td><td>1024 kbps</td></tr>
</table>

Figure 6-29 CIF - High texture Low motion - H.264/25 fps

		
Original		
		
30 fps 256 kbps	30 fps 512 kbps	30 fps 1024 kbps

Figure 6-30CIF - High texture Low motion - H.264/30 fps

6.2.2.3 Low Texture High Motion

MOS results of this video sequence type are shown in Table 6-8 and Figure 6-31. At 256 kbps, low frame rates produced the highest quality results. MOS at 15 and 20 fps was 2.66.

This degraded to become the worst at 25 fps with a value of 2 and its average at 30 fps was 2.33.

At 512 kbps, it is clear that increasing the frame rate worsened the quality, where the highest MOS was at 15 fps and decreased to 2.66 at 30 fps. Unfortunately, increasing the bit rate from 256 kbps to 512 and increasing the frame rate from 15 to 30 fps had no usefulness as MOS did not increase. This means that the bandwidth was additionally exhausted with no noticeable increase in video quality.

Bit Rate (kbps)	Frame Rate (fps)	MOS
256	15	2.66
	20	2.60
	25	2
	30	2.33
512	15	3.33
	20	3
	25	2.66
	30	2.66
1024	15	3
	20	3
	25	3
	30	3.33

Table 6-8 CIF – Low texture High motion sequences – H.264/DSIS

At the higher bit rate of 1024 kbps, increasing frame rate from 15 to 25 fps had no effect on quality and MOS was similar. The higher frame rate at 30 fps had little effect on quality increase compared to lower frame rates.

As shown in Figure 6-32, Figure 6-33, Figure 6-34 and Figure 6-35, snapshots of the different frame rates and bit rates have been captured to make clear comparison of compression results.

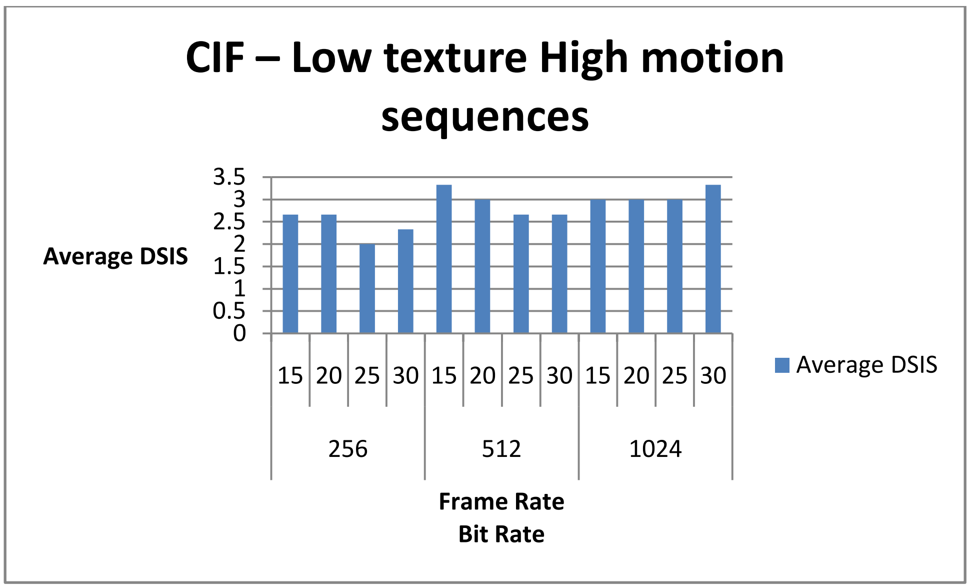

Figure 6-31 CIF – Low texture High motion sequences – H.264/DSIS

		
Original		
		
256 kbps	512 kbps	1024 kbps

Figure 6-32 CIF - Low texture High motion - H.264/15 fps

<table>
<tr><td colspan="3">
</td></tr>
<tr><td colspan="3">Original</td></tr>
<tr><td>
</td><td>
</td><td>
</td></tr>
<tr><td>256 kbps</td><td>512 kbps</td><td>1024 kbps</td></tr>
</table>

Figure 6-33 CIF - Low texture High motion - H.264/20 fps

Original

		
256 kbps	512 kbps	1024 kbps

Figure 6-34 CIF - Low texture High motion - H.264/25 fps

		
Original		
		
30 fps 256 kbps	30 fps 512 kbps	30 fps 1024 kbps

Figure 6-35 CIF - Low texture High motion - H.264/30 fps

6.2.2.4 Low Texture Low Motion

As demonstrated in Table 6-9 and Figure 6-36, the MOS results slightly increased at higher bit rates. Although 512 kbps and 1024 kbps produced comparable results, 256 kbps resulted in less quality than these two bit rates. At 256 kbps, MOS produced the highest score at 15fps. The remaining frame rates produced similar quality evaluations.

As expected, higher bit rates showed better MOS than the lower ones. At 512 kbps, the frame rates 15 and 30 resulted in higher MOS compared to 20 and 25. As a result, increasing frame rate from 15 to 30 at 512 kbps was not useful as MOS did not change. With this group, again MOS results at 1024 kbps produced similar scores with 512 kbps.

Bit Rate (kbps)	Frame Rate (fps)	MOS
256	15	3.33
	20	2.33
	25	2.33
	30	2.33
512	15	3.2
	20	3
	25	3
	30	3.33
1024	15	3
	20	3.33
	25	3.66
	30	3.33

Table 6-9 CIF – Low texture Low motion sequences – H.264/DSIS

According to the subjective evaluation, only 25 fps at 1024 kbps outperformed the 512 kbps results. On average, there seems no perceptible superiority between the two bit rates.

As a consequence of the results above, increasing the bit rate from 256 kbps to 1024 kbps and frame rate from 15 fps to 30 fps had no useful effect. The MOS results showed similarity for both of them. Therefore, it is a waste of bandwidth to increase them when the quality does not improve.

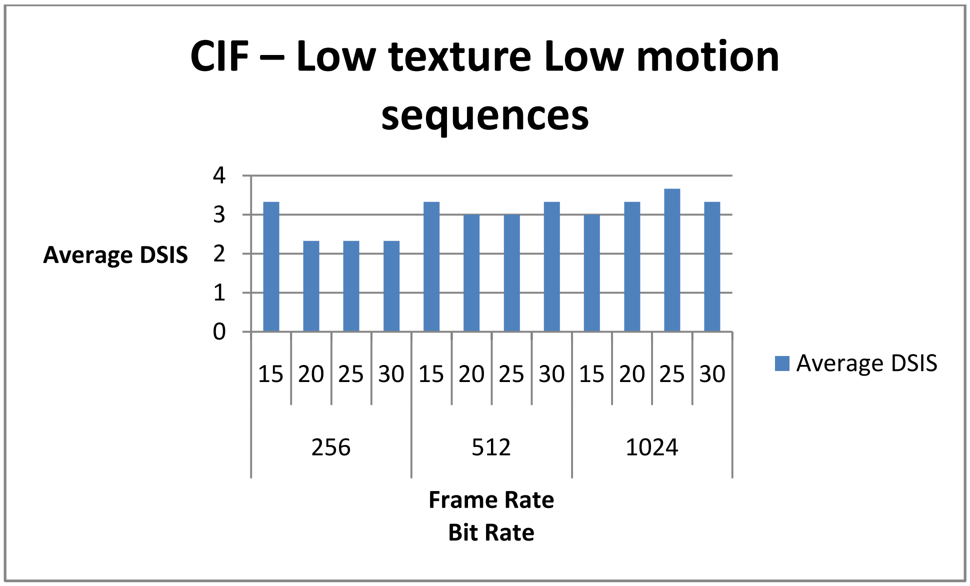

Figure 6-36 CIF – Low texture Low motion sequences – H.264/DSIS

The snapshots of the compressed and original video sequences are illustrated in Figure 6-37, Figure 6-38, Figure 6-39 and Figure 6-40.

<table>
<tr><td colspan="3">
</td></tr>
<tr><td colspan="3">Original</td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td>256 kbps</td><td>512 kbps</td><td>1024 kbps</td></tr>
</table>

Figure 6-37 CIF - Low texture Low motion - H.264/15 fps

<table>
<tr><td colspan="3">
</td></tr>
<tr><td colspan="3">Original</td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td>256 kbps</td><td>512 kbps</td><td>1024 kbps</td></tr>
</table>

Figure 6-38 CIF - Low texture Low motion - H.264/20 fps

Original

		
256 kbps	512 kbps	1024 kbps

Figure 6-39 CIF - Low texture Low motion - H.264/25 fps

<table>
<tr><td colspan="3">
</td></tr>
<tr><td colspan="3">Original</td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td>256 kbps</td><td>512 kbps</td><td>1024 kbps</td></tr>
</table>

Figure 6-40 CIF - Low texture Low motion - H.264/30 fps

6.2.3 QCIF Format Resolution

6.2.3.1 High Texture High Motion

As shown in Table 6-10 and Figure 6-41, MOS results improved slightly as the bit rate was increased. At 256 kbps, MOS was, on average, 1.41 whereas it was 1.57 at 512 and 1024 kbps. Increasing the bit rate from 512 kbps to 1024 kbps in this video type appeared to have no visible effect.

Bit Rate (kbps)	Frame Rate (fps)	MOS
256	15	1.33
	20	1.33
	25	1.33
	30	1.66
512	15	1.66
	20	1.66
	25	1.33
	30	1.66
1024	15	1.33
	20	1.66
	25	1.66
	30	1.66

Table 6-10 QCIF – High texture High motion sequences – H.264/DSIS

At the lower bit rate of 256 kbps, the quality did not change with the frame rate increase from 15 to 20 or 25 fps. The only change occurred slightly at 30 fps. This was not the same with the bit rate of 512 kbps, where all frame rates produced the same MOS result except for 25 fps where quality slightly degraded.

At the higher bit rate of 1024 kbps, MOS was the lowest at 15 fps and increased to be a little better at other higher frame rates. It is notable that increasing frame rate from 20 to 25 or 30 at 1024 kbps was not useful as the quality did not improve. Furthermore, MOS at 256 kbps and 30 fps was equal to MOS at 1024 kbps and 30 fps. In conclusion, it is a waste of broadband to increase bit rate in such a way when the quality is not getting better.

All the compressed video sequences were snapshot and the snapshots are shown in Figure 6-42, Figure 6-43, Figure 6-44 and Figure 6-45. The snapshots show insignificant differences for the different frame rates.

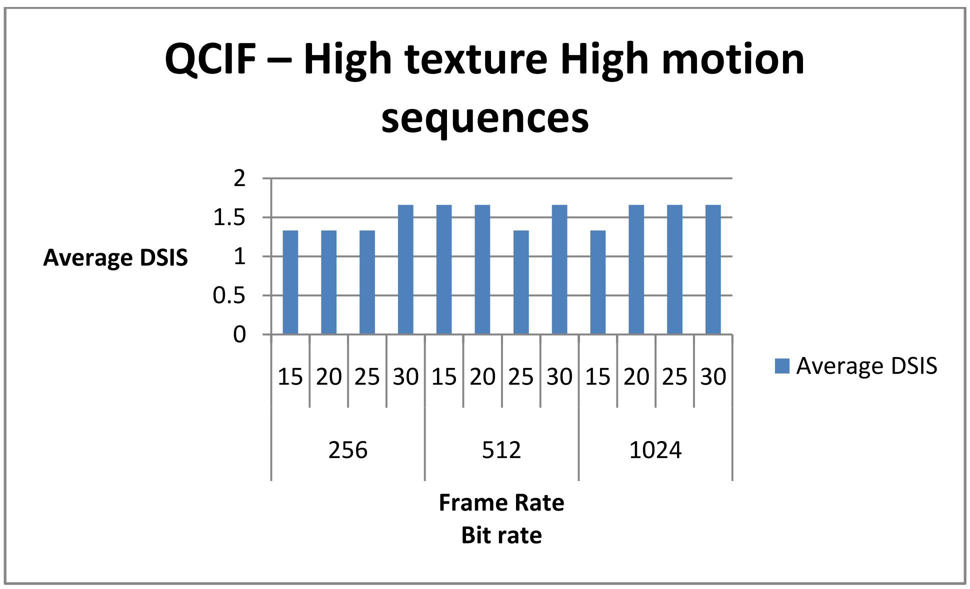

Figure 6-41 QCIF – High texture High motion sequences – H.264/DSIS

		
Original		
		
256 kbps	512 kbps	1024 kbps

Figure 6-42 QCIF - High texture High motion - H.264/15 fps

<table>
<tr><td colspan="3">
</td></tr>
<tr><td colspan="3">Original</td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td>256 kbps</td><td>512 kbps</td><td>1024 kbps</td></tr>
</table>

Figure 6-43 QCIF - High texture High motion - H.264/20 fps

		
Original		
		
256 kbps	512 kbps	1024 kbps

Figure 6-44 QCIF - High texture High motion - H.264/25 fps

Original

256 kbps

512 kbps

1024 kbps

Figure 6-45 QCIF - High texture High motion - H.264/30 fps

6.2.3.2 High Texture Low Motion

The MOS results of this video sequence type are illustrated in Table 6-11 and Figure 6-46. At the bit rate of 256 kbps, the quality increased as the frame rate was increased except for the quality at 15 and 20 fps which were similar. MOS increment between 20 and 25 fps was not large and was not worth the bandwidth consumption made by the frame rate increase. At 30fps, the quality increased up to 1.2 with an average MOS enhancement of 0.2.

Bit Rate (kbps)	Frame Rate (fps)	MOS
256	15	1
	20	1
	25	1.06
	30	1.2
512	15	1
	20	1
	25	1.06
	30	1.13
1024	15	1
	20	1.33
	25	1
	30	1.33

Table 6-11 QCIF – High texture Low motion sequences – H.264/DSIS

At 512 kbps, the results appeared similar except that the frame rate of 30 fps was found to be of less quality. The first three frame rates between 256 and 512 kbps were equal and the bit rate increase did not improve the quality. As a result, increasing bit rate from 256 kbps to 512 kbps worsened the quality.

At the higher bit rate of 1024 kbps, the MOS did not improve significantly. The only improvement was at 20 and 30 fps. Conversely, MOS at 15 fps and 1024 kbps was similar to the result at 15 fps with lower bit rates. As a conclusion, increasing bit rate at 15 fps and 20 fps did not improve the quality and may cause some quality degradation.

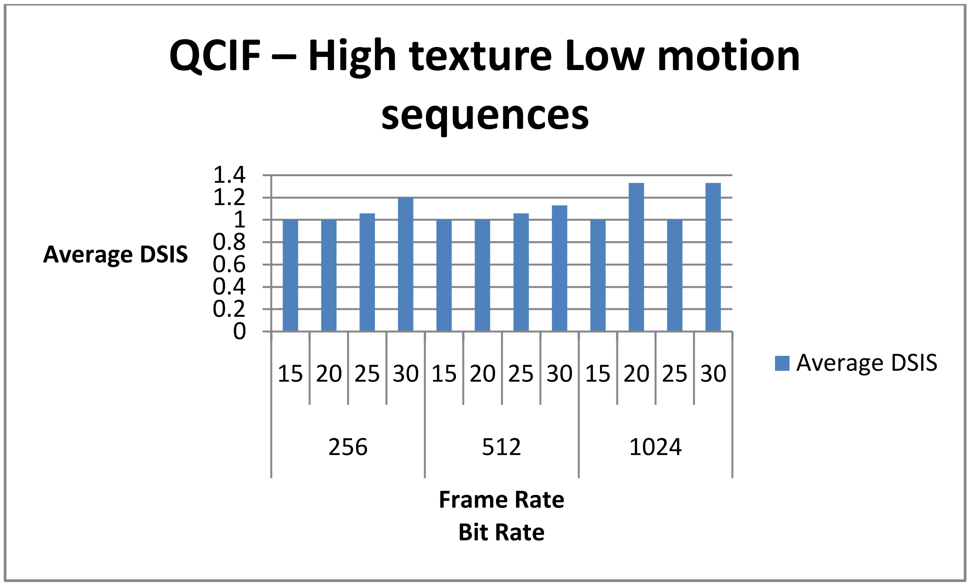

Figure 6-46 QCIF – High texture Low motion sequences – H.264/DSIS

The snapshots of this video sequence at the different bit rates and frame rates are shown in Figure 6-47, Figure 6-48, Figure 6-49 and Figure 6-50.

Original

256 kbps	512 kbps	1024 kbps
		

Figure 6-47 QCIF - High texture Low motion - H.264/15 fps

<table>
<tr><td colspan="3">
</td></tr>
<tr><td colspan="3">Original</td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td>256 kbps</td><td>512 kbps</td><td>1024 kbps</td></tr>
</table>

Figure 6-48 QCIF - High texture Low motion - H.264/20 fps

<table>
<tr><td colspan="3">
</td></tr>
<tr><td colspan="3">Original</td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td>256 kbps</td><td>512 kbps</td><td>1024 kbps</td></tr>
</table>

Figure 6-49 QCIF - High texture Low motion - H.264/25 fps

Original

256 kbps	512 kbps	1024 kbps
		

Figure 6-50 QCIF - High texture Low motion - H.264/30 fps

6.2.3.3 Low Texture High Motion

Table 6-12 and Figure 6-51show the results of this type of video sequences. At the 256 kbps bit rate, the MOS result was the worst at 15 fps and the best result was at 20 fps.

However, with this bit rate with higher frame rates of 25 and 30 fps, the MOS result was better than 15 but worse than 20 on average. As a result, the best MOS result at the 256 kbps bit rate was at the moderate frame rate of 20 fps since higher or lower frame rates resulted in less quality.

Bit Rate (kbps)	Frame Rate (fps)	MOS
256	15	1.33
	20	2
	25	1.66
	30	1.66
512	15	2
	20	2
	25	2
	30	1.66
1024	15	1.66
	20	1.66
	25	1.66
	30	1.66

Table 6-12 QCIF – Low texture High motion sequences – H.264/DSIS

The MOS results at higher bit rates were steadier than at the lower. For the bit rate of 512 kbps, all frame rates produced the same quality evaluation of 2 except for the quality at 30 fps which slightly degraded and registered 1.66. Therefore, increasing the frame rate from

15 to 20 or 25 was useless and, furthermore, it worsened MOS when the frame rate was increased to 30 fps. The MOS result at 1024 kbps was stable. It is clear that changing the frame rate had no effect on the quality. All frame rates had the same MOS score of 1.66.

As a conclusion, the best MOS results of this type of video sequences were at either the lower bit rate of 256 kbps and 20 fps, or at 512 kbps but with low frame rates between 15 and 25 fps. Nevertheless, the 256 kbps bit rate with 20 fps is the recommendation here as the 512 kbps consumes more of the bandwidth.

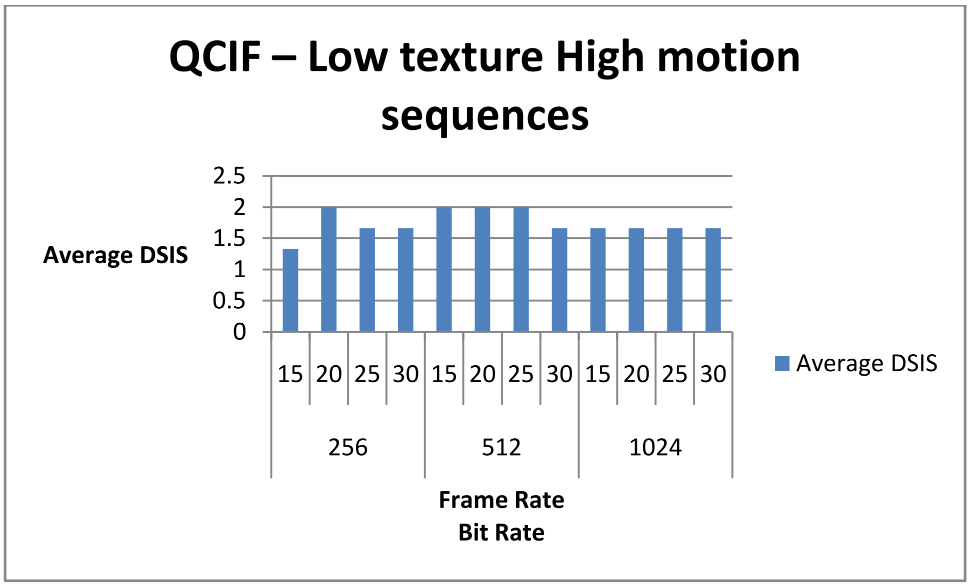

Figure 6-51 QCIF – Low texture High motion sequences – H.264/DSIS

The snapshots in Figure 6-52, Figure 6-53, Figure 6-54 and Figure 6-55 show the effects of compressing this type of video sequences in different scenarios of frame rates and bit rates.

<table>
<tr><td colspan="3">
</td></tr>
<tr><td colspan="3">Original</td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td>256 kbps</td><td>512 kbps</td><td>1024 kbps</td></tr>
</table>

Figure 6-52 QCIF - Low texture High motion - H.264/15 fps

<table>
<tr><td colspan="3">
</td></tr>
<tr><td colspan="3">Original</td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td>256 kbps</td><td>512 kbps</td><td>1024 kbps</td></tr>
</table>

Figure 6-53 QCIF - Low texture High motion - H.264/20 fps

<table>
<tr><td colspan="3">
</td></tr>
<tr><td colspan="3">Original</td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td>256 kbps</td><td>512 kbps</td><td>1024 kbps</td></tr>
</table>

Figure 6-54 QCIF - Low texture High motion - H.264/25 fps

		
Original		
		
256 kbps	512 kbps	1024 kbps

Figure 6-55 QCIF - Low texture High motion - H.264/30 fps

6.2.3.4 Low Texture Low Motion

MOS results of this type of video sequences are illustrated in Table 6-13 and Figure 6-56. Results show that the lowest MOS was at 256 kbps with 15 fps and at 512 kbps with 15 and 25 fps. At 256 kbps, increasing the frame rate from 20 to 25 or 30 did not increase the quality. On the other hand, MOS results of bit rate 512 kbps were the best at 20 and 30 fps. Thus, increasing bit rate from 256 kbps to 512 kbps was not useful and is not recommended since this will consume more bandwidth with no quality improvement.

Bit Rate (kbps)	Frame Rate (fps)	MOS
256	15	1.66
	20	2
	25	2
	30	2
512	15	1.66
	20	2
	25	1.66
	30	2
1024	15	2
	20	2
	25	2
	30	2

Table 6-13 QCIF – Low texture Low motion sequences – H.264/DSIS

At the 1024 kbps bit rate, the MOS quality was similar for all frame rates. The increase of frame rate did not affect the quality. Since frame rate increase normally consumes bandwidth, it is useless to increase it while the quality does not improve. Moreover, good MOS results at 1024 kbps and 30 fps can be obtained at lower bit rates and frame rates. This was clear at 256 kbps and 20 fps. In conclusion, this huge bit rate and frame rate increment is considered as bandwidth wasted with no benefits to quality improvement.

The snapshots of the compressed and original video sequences are shown in Figure 6-57, Figure 6-58, Figure 6-59 and Figure 6-60. The images demonstrate that the quality differences among the various bit rates and frame rates are unnoticeable.

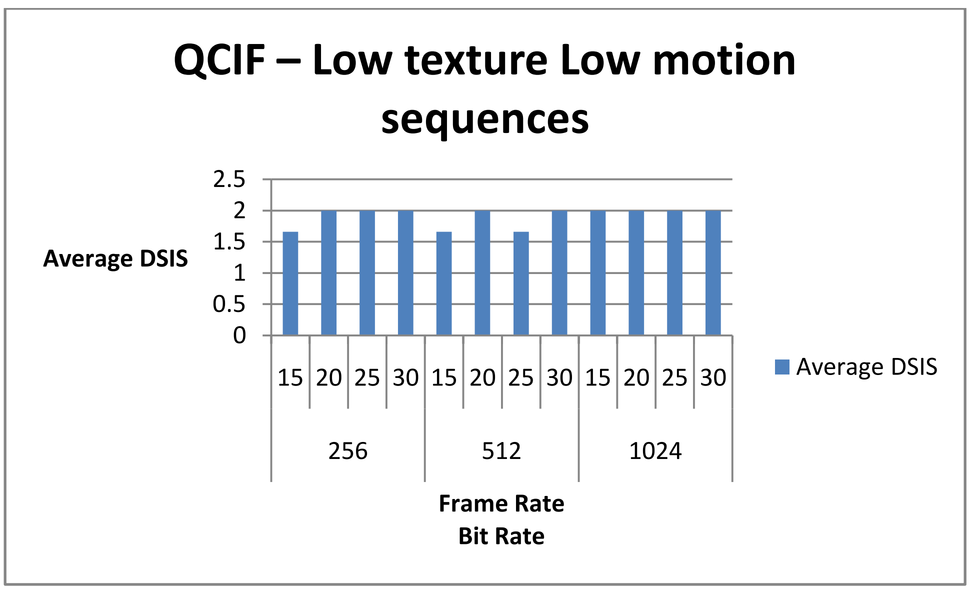

Figure 6-56 QCIF – Low texture Low motion sequences – H.264/DSIS

<table>
<tr><td colspan="3">
</td></tr>
<tr><td colspan="3">Original</td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td>256 kbps</td><td>512 kbps</td><td>1024 kbps</td></tr>
</table>

Figure 6-57 QCIF - Low texture Low motion - H.264/15 fps

Original

		
256 kbps	512 kbps	1024 kbps

Figure 6-58 QCIF - Low texture Low motion - H.264/20 fps

<table>
<tr><td colspan="3">
</td></tr>
<tr><td colspan="3">Original</td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td>256 kbps</td><td>512 kbps</td><td>1024 kbps</td></tr>
</table>

Figure 6-59 QCIF - Low texture Low motion - H.264/25 fps

<table>
<tr><td colspan="3">
</td></tr>
<tr><td colspan="3">Original</td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td>30 fps 256 kbps</td><td>30 fps 512 kbps</td><td>30 fps 1024 kbps</td></tr>
</table>

Figure 6-60 QCIF - Low texture Low motion - H.264/30 fps

6.3 STATISTICAL STUDY OF SUBJECTIVE TESTING RESULTS

The comparison study between video codecs was completed in the preceding chapter. Results showed that H.264 video codec outperforms other codecs in subjective and objective tests. Based on the results in the chapter, H.264 is chosen as the most recommended codec for synthetic video sequences.

This chapter performs a methodical analysis to study the influence of different dimensions on MOS. The study considers a wide scope of dimensions that affects the perceptual visual quality. In preceding sections, the results of subjective view tests for assessing the perceptual quality achieved were presented. MOS results were shown in details for different Bit rate, Frame rate and Frame size.

This chapter presents an Analysis of Variance (ANOVA) statistical study of a MOS data set to determine how MOS is affected by video type (VT), bit rate (BR) and frames per second (Fps) for different frame sizes (FS). A three-way ANOVA is adopted in these tests.

The analysis focuses on capturing any difference in MOS mean under the potential factors above. The model that will be tested is as follows:

$$MOS = BR + FpS + VT + \varepsilon \qquad (6-1)$$

In the model illustrated in (6-1), it should be noticed that the error is assumed normally and identically distributed with mean 0 and varianceσ^2. As discussed in Section 4.3, the three assumptions for ANOVA should be validated before applying the analysis; all assumptions are validated where these assumptions are:

1- Normality assumption: We use large sample theory to justify the normality of our sample since the sample size is over 30.
2- Independency: The independency assumption is satisfied since no MOS result is affected by the others.
3- Homoscedasticity (equality of variances): The Levene's Test of Equality of Error Variances is conducted and the results showed that no major violation of this

assumption is observed. This is because the p-value is greater than 0.05, and is considered as insignificant.

The results of ANOVA analysis of QCIF is shown in Table 6-14. Each column shows a specific output and can be explained as follows: the second column is the sum of squares between treatments of each dimension. The third column is the degrees of freedom related with each dimension model; it can be calculated as (number of treatments – 1). The fourth column shows the ratio of sum of squares to degrees of freedom. The fifth is the F statistic, and the last is the p-value. The p-value indicates the significance of the dimension and can be derived from the cumulative distribution function of F.

As shown in Table 6-14, the p-value is less than 0.05 for both Fps and VT where it is greater for BR. This indicates that MOS are affected by both Fps and VT significantly. However, the BR is not considered to be affecting MOS as the corresponding p-value is insignificant ($p>=0.05$).

As the p-value of FpS and VT are ($p<=0.05$), this indicates that these two dimensions substantially affect MOS. However, the small p-value of VT which is almost zero implies that MOS is severely affected by this dimension, where MOS is affected slightly less by Fps.

Source	Sum of Squares	Degrees of Freedom	Mean Square	F statistic	p-value
BR	.030	2	.015	.602	.553
FpS	.241	3	.080	3.228	.033
VT	4.562	3	1.521	61.193	.000

Table 6-14 Three-Way ANOVA on MOS of QCIF

The pairwise comparison study of frame rate at QCIF resolution is illustrated in Table 6-15. As shown in the second column of the table, the mean difference magnitude indicates that

Fps represented by (I) affects on MOS better than (J) if it is positive and worse if negative. The results show that increasing frame rate from 15 fps into higher rates has increased the MOS, but it is clear from the p-value, that increasing frame rate is the most significant at 20 fps. Increasing it more than 20 fps is less significant, and this means increasing frame rate more than 20 fps will have worse effects on the MOS in addition to the bandwidth consumption.

(I) FpS	(J) FpS	Mean Difference (I-J)	Std. Error	p-value	95% Confidence Interval for Difference	
					Lower Bound	Upper Bound
15.00	20.00	-.167	.064	.013	-.298	-.037
	25.00	-.066	.064	.313	-.196	.064
	30.00	-.166	.064	.014	-.296	-.036
20.00	15.00	.167	.064	.013	.037	.298
	25.00	.102	.064	.122	-.029	.232
	30.00	.002	.064	.979	-.129	.132
25.00	15.00	.066	.064	.313	-.064	.196
	20.00	-.102	.064	.122	-.232	.029
	30.00	-.100	.064	.128	-.230	.030
30.00	15.00	.166	.064	.014	.036	.296
	20.00	-.002	.064	.979	-.132	.129
	25.00	.100	.064	.128	-.030	.230

Table 6-15 Pairwise Comparison of Frame-rate at QCIF

As a result, at such low spatial resolution it is not always true that increases in the frame rate lead to increasing the perceptual quality. This claim is clearly shown by the results of comparing 15 fps to the other frame rates. The Mean Difference values show that 20 fps at QCIF resolution produces better MOS than higher frame rates.

In the case of comparing 30 fps to other frame rates, results show that increasing frame rate from 15 to 30 is significant ($p<=0.05$) and highly affects MOS, where it is insignificant at 20 and 25 fps. The full pairwise comparisons are shown in Table 6-15.

Results of CIF resolutions are illustrated in Table 6-16. Results show that the p-value is border significant for VT, while it is insignificant for BR and Fps. This implies that MOS in CIF resolution is slightly affected by VT. However, the effect of BR and Fps on MOS is negligible.

Source	Sum of Squares	Degrees of Freedom	Mean Square	F statistic	p-value
VT	2.252	3	.751	2.771	.054
BR	.517	2	.258	.954	.394
FpS	.173	3	.058	.213	.887

Table 6-16 Three-Way ANOVA on MOS of CIF

Comparing the magnitude of the p-value of BR and Fps, the p-value 0.394 of BR is closer to the 0.05 the significance threshold, where Fps with 0.887 is less close. This indicates that BR could affect the MOS more than Fps in a case of considering their reformatting.

The pairwise comparison of CIF resolution is illustrated in Table 6-17. Results show that increasing frame rates from 15 to 20, 25 and 30 fps is insignificant (p-value>=0.05). In addition, and as shown in the results, increasing frame rate from 15 to 20 or 25fps will affect negatively on the perceptual quality. As shown in the comparison of 30fps to other frame rates, increasing the frame rate to 30 fps enhances the perceptual quality. However, this enhancement is considered insignificant due to viewer's subjective reaction.

(I) FpS	(J) FpS	Mean Difference (I-J)	Std. Error	p-value	95% Confidence Interval for Difference	
					Lower Bound	Upper Bound
15.00	20.00	.001	.212	.997	-.429	.431
	25.00	.111	.212	.605	-.319	.541
	30.00	-.054	.212	.800	-.484	.376
20.00	15.00	-.001	.212	.997	-.431	.429
	25.00	.110	.212	.608	-.320	.540
	30.00	-.055	.212	.797	-.485	.375
25.00	15.00	-.111	.212	.605	-.541	.319
	20.00	-.110	.212	.608	-.540	.320
	30.00	-.165	.212	.442	-.595	.265
30.00	15.00	.054	.212	.800	-.376	.484
	20.00	.055	.212	.797	-.375	.485
	25.00	.165	.212	.442	-.265	.595

Table 6-17 Pairwise Comparison of Frame-rate at CIF

Table 6-18 shows the results of 4CIF resolution. As can be seen from the p-values, all the dimensions are insignificant. The results indicate that the VT, BR and FpS are affected

MOS insignificantly. Nevertheless, according to the magnitude of p-values, we can claim that BR affects the MOS result the most, the FpS affects less and the least effect is of the VT.

It can be concluded from Table 6-18 that 4CIF video sequences are less affected by VT. Furthermore, the VT has the least effect on such resolutions. The effect of VT increases at CIF and QCIF resolutions. On the other hand, the BR effect is increased as the frame size is increased, BR shows higher effect at CIF and the highest effect is at 4CIF.

Source	Sum of Squares	Degrees of Freedom	Mean Square	F statistic	p-value
VT	1.824	3	.608	.428	.734
BR	4.731	2	2.365	1.664	.203
FpS	3.073	3	1.024	.720	.546

Table 6-18 Three-Way ANOVA on MOS of 4CIF

Table 6-19 shows the pairwise comparison of frame rates at 4CIF resolution. It is clear that the higher frame rate increases the perceptual quality. However, the increase is considered insignificant as ($p>=0.05$). The results show that any increase of frame rate will affect positively on MOS. Mean Difference results show that higher frame rates provide higher MOS. The detailed comparison is provided in the table.

The previous result analysis studied the effects of each element on the resolution and on the MOS results. Another important contribution of this analysis is to present the recommended tradeoffs between frame rate and frame resolution for given bit rates. The recommendations for each bit rate over different video types are given in Table 6-20.

(I) FpS	(J) FpS	Mean Difference (I-J)	Std. Error	p-value	95% Confidence Interval for Difference	
					Lower Bound	Upper Bound
15.00	20.00	.390	.487	.428	-.595	1.375
	25.00	.546	.487	.269	-.439	1.530
	30.00	.673	.487	.174	-.311	1.658
20.00	15.00	-.390	.487	.428	-1.375	.595
	25.00	.156	.487	.751	-.829	1.140
	30.00	.283	.487	.564	-.701	1.268
25.00	15.00	-.546	.487	.269	-1.530	.439
	20.00	-.156	.487	.751	-1.140	.829
	30.00	.128	.487	.795	-.857	1.112
30.00	15.00	-.673	.487	.174	-1.658	.311
	20.00	-.283	.487	.564	-1.268	.701
	25.00	-.128	.487	.795	-1.112	.857

Table 6-19 Pairwise Comparison of Frame-rate at 4CIF

Table 6-20 shows the recommended tradeoffs for different bit rates and video types. The bit rate is presented in kilobit per second (kbps). Video types shown in the table are high texture high motion (HTHM), high texture low motion (HTLM), low texture high motion (LTHM) and low texture low motion (LTLM) videos. As shown in the table, each bit rate has its own best combinations of frame rate and frame resolution. These combinations can vary when changing video sequence type and bit rate.

Bit rate/kbps	HTHM	HTLM	LTHM	LTLM
256	4CIF - 15 fps	4CIF – 15fps	CIF 15fps and 4CIF-15fps	CIF–15
512	CIF - 20 fps and 4CIF-20 fps	4CIF-15 fps	4CIF-15 and 20 fps	CIF-30 fps and 4CIF-30 fps
1024	4CIF - 25, 30 fps	4CIF - 15 fps	4CIF - 15 and 20 fps	4CIF-15 fps

Table 6-20 Recommended Tradeoffs

7 CONCLUSION AND FUTURE WORK

7.1 Conclusion of the Study

In this thesis, an on-line game video streaming system has been proposed. The main idea is to combine fast compression, high efficiency coding, bandwidth exploiting, best user satisfaction and automatic tradeoff detection.

It has been shown in the thesis that on-line game video streaming services can be developed to provide higher user satisfaction with the same available bandwidth and common video coding techniques. The proposed system can be used in practice to supply all on-line game services with greater efficiency. In addition, the thesis presents several new contributions.

The first contribution is an in-depth comparative study between the most common video coding techniques; MPEG-4 Visual, H.264 and VP-8. The H.265 video codec is also known as High Efficiency Video Coding (HEVC). It is a draft descendant of H.264 developed to significantly reduce the compression size by approximately 50% compared to existing standards [96]. In addition, HEVC contains many new technical aspects which make significant enhancements such as the new block partitioning structure [97]. However, H.265 was not used in this study because it is still under development by ISO MPEG and ITU-T VCEG, and it will not be released until early 2013.

The coding techniques were evaluated using PSNR as an objective metric and DSIS for subjective evaluation. The effect of codec type, frame rate and frame resolution were considered in the evaluation tests. In addition, several bit rates were used to present a wider view for evaluation.

The compression experiments showed that H.264 often produced the highest evaluation among the codecs for both subjective and objective metrics. In addition, MPEG-4 had the best compression performance at lower bit rates, CIF resolution and high texture and motion sequences. The obtained subjective results were statistically analyzed, and the influence of the different dimensions on the MOS has been illustrated.

The second contribution presented a fast inter mode selection for H.264. An early termination process with reference frame management technique has been developed. The proposed method achieved significant time reduction with negligible loss of quality. The study considered various video sequences which included different frame rates, resolutions and visual content.

The proposed algorithm achieved encoding time savings of at least 53.29% for IPPP and 54.83% for IBBP for the test sequences. The higher gain of time reduction with the small loss of coding efficiency makes the proposed algorithm more suitable for fast inter mode coding especially for applications that consider total encoding time.

The third contribution is a tradeoff study between frame rate and resolution within specific bandwidths. The study had been investigating the most preferable compromise between these two important factors according to perceptual user evaluation. Recommendations on the tradeoff results are presented to achieve users' satisfaction.

The study showed that 4CIF video sequences are less affected by the video type (VT). Furthermore, the VT has the least effect on such resolutions. The effect of VT increases at CIF and QCIF resolutions. On the other hand, the bit rate (BR) effect is increased as the frame size is increased, BR shows higher effect at CIF and the highest effect is at 4CIF. The analysis studied the effects of each element on the resolution and on the MOS results. The analysis also presented the recommended tradeoffs between frame rate and frame resolution for given bit rates.

The thesis presented several worthy contributions in the field of on-line game video streaming. Results showed that frame rate and frame resolution can significantly affect compression efficiency and user satisfaction. Also, video codecs respond to the change of frame rate and resolution dissimilarly. Another important outcome shows that increasing frame rate or frame size will not always guarantee higher user satisfaction.

The high compression complexity of on-line systems is also a problem that has been studied in this thesis. The proposed fast mode selection algorithm presented a new model to speed up compression which will improve on-line game streaming experience.

The most important factor of on-line game designer is to adopt new techniques that attract game players. The results reported here do this in numerous ways. This study also presented recommended tradeoffs between frame rates and frame resolution within given bitrates to present highest user satisfaction by exploiting the available uplink capabilities.

7.2 Future Work

7.2.1 Introduction

A proposed model should provide the main services provided by other game server browsers systems where the most common features are:

1. Game detecting: This allows user detecting other players who are sharing their games and allows contacting them. This provides information about current games being played, how many hours the game has been played, servers running the games and scores of players.
2. Game sequence capturing and recording: While the game is being played, video of the game needs to be captured to allow transmission and streaming. In addition, sometimes users need to record some scenes and snapshots of games while it is running for their own records. The process of capturing and recording has a significant impact on game performance. In low performance systems the frame rate is decreased severely to allow the recording process. This problem is common for all in-game recording and not unique to specific game browsers.
3. Sequence compression: Before the process of streaming, captured sequences have a huge amount of data. These sequences need to be compressed and then streamed.

4. Live streaming: The captured and compressed scenes to be viewable by other users using a web browser plug-in.

7.2.2 Proposed Model

As the main contribution of this thesis is to provide a complete video games streaming system, in Chapter 6, recommendations are presented for the tradeoff between frame rate and resolution according to perceptual user evaluation. Therefore, to manage the process of game compression and streaming, a modified model is proposed as an enhanced web game streaming service. The proposed model will provide several services and techniques that are put into operation in some phases. The proposed model can be summarized as follows:

Model Phases:

1. Game detecting.
2. Game sequence capturing.
3. Video segmentations.
4. Video classification.
5. Compression specifications selection.
6. Compression process.
7. Streaming process.

According to the proposed model, the games being played are detected by first using their information on the web game browser. Fraps software [88] was used for this study. Once the gamer chooses a game to play and launches it on his PC, the client software starts capturing the displayed sequence of the game.

The captured sequence needs to be specified under which contents and motion is classified. Many video classification techniques are proposed in the literature. Techniques of texture and motion classification need to be low computational cost methods for such online compression systems.

The video classification focuses on the video content material (texture) and content motion. Classifying according to the motion can be categorized into two main categories. First is a major motion which studies camera motion and scene change. Second is minor motion and studies objects' motion, camera shaking and zooming. On the other hand, testing the texture studies distribution of colors in a video frame which is often represented using a color histogram. That is, a count of how many pixels in the frame exist for each possible color. Color histograms are often used for comparing two frames with the assumption that similar frames will have similar counts.

In our proposed model, and before the classification step, a video sequence should be segmented in equal slice length. The proposed slice length is 10 seconds. The length of these slices guarantees that the change of the sequence will be considered each ten seconds.

However, shortening the slice length is expected to lead to high computational processing. By this segmentation, the highest benefit of compression tradeoff recommendations can be exploited. The classification will determine that the video slice is classified under one of the four categories: High texture high motion (HTHM), high texture low motion (HTLM), low texture high motion (LTHM) and low texture low motion (LTLM). Each segment is compressed using the recommended parameters that have been gained in Chapter 5. The compressed slices are streamed over the game web browser.

Model Diagram:

For completeness, the model diagram is shown in **Ошибка! Источник ссылки не найден.**. After the model platform is launched, it detects what games are advertised on the game server and compatible with its software. This step is conducted first and referred to in the figure as game detection.

In the second step, the user needs to determine which game her or she intends to stream and which server to use. Then, the user starts the game and the system starts real-time capturing in the Game Capturing step. The game is directly classified as high definition (HD) or not. In case of HD, there is no need to check the sequence texture as it is considered a high texture sequence.

The system starts slicing the sequence and checks to classify the slice as high motion (HM) or low motion (LM). On the other hand, if the captured video is not HD it should be sliced; and the slice is tested for its texture and motion. A process is conducted to classify the slice in one of the categories HTHM, HTLM, LTHM and LTLM and the recommended tradeoff is used for compression.

Finally, the recommended tradeoff between frame rate and resolution is chosen to compress the slice before it is streamed to other players.

Figure 7-1 Proposed On-line Game Service Model

A complete system components diagram is shown in **Ошибка! Источник ссылки не найден.**. The proposed model in **Ошибка! Источник ссылки не найден.** is mentioned in this complete system as the Interface Platform. The complete system can be explained from the top down as follows:

1- The system contains three main phases: Client side, HTTP phase and Game servers' side.

2- On the client side, gamers can launch the game using the client game installed on their PCs. Gamers can also set their setting and view all parameters related to the game itself (Avatar, game environment, etc.).

3- The interface platform can be installed from game servers containing our proposed model.

4- The gamers launch the game and request connecting to the desired servers advertising available games.

5- The servers respond to the user request using all related personal data that are stored on the database.

6- The gamers start playing and the proposed model starts as well. The model starts auto-capturing of running game, classifying the video game each ten seconds, choosing the appropriate compression technique and finally streaming the sequence.

7- Video classification is not a new research field. Many researches and methods have been presented on this area [98]. Automatic video classification features are categorized in three main categories: text, audio and visual. A variety of feature combinations is also used in the automatic video classification.

8- Many visual classification techniques are presented. One technique is based on color features where classification can be conducted using color distribution. This distribution is often presented using color histograms. Another technique is based on shot features where different mechanisms are used [99-101]. A classification techniques based on objects is proposed but still uncommon because of the high computational cost and difficulty of detecting [98]. Some researchers used the idea of object features for specific object types [102-104]. One other visual classification involves motion-based features. Here, motion in a video is categorized as movement of the objects in a scene and movement of camera actions [105-107].

9- Since people receive much of the information through the sense of vision, this research classifies video sequences according to their content (texture) and objects' speed (motion speed). However, fast and efficient classification techniques to classify an input sequence of the proposed system are required. The selected techniques should take into consideration the requirements of this live streaming system.

10- The proposed model requires team work to implement it. A network engineer, web programmer and software engineer would be essential for a successful implementation.

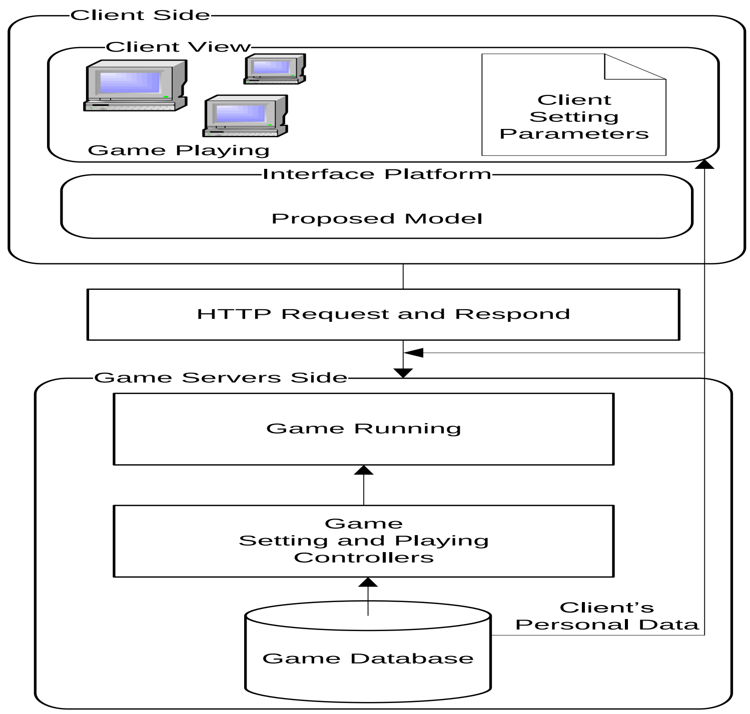

Figure 7-2 Complete On-line Game Service System Component

7.3 Limitations

The proposed system was evaluated for three video coding techniques, VP8, MPEG-4 Visual and H.264. For completeness, other codecs such as HEVC and VP9 could be tested. In addition, the experiments were done using the World of Warcraft game as it is one of the most used public games. Other games with different visual and motion elements could be considered in future works.

To evaluate video quality and perceptual results PSNR and DSIS were used. However other quality metrics could be used in the future for a more comprehensive evaluation.

The proposed fast inter mode selection algorithm provided significant savings of total encoding time. However, this algorithm can be combined with a fast intra mode selection technique to give further encoding time reduction.

The tradeoff between frame rate and resolution was presented in this research. A tradeoff considering time complexity is expected to present valuable new outcomes.

The proposed on-line game service model is suitable for PC computers. Research of similar systems for handheld devices is expected to be a successful research route.

Video classification is not a new research field, but it is possible to search for fast techniques that could provide more sensitive and more applicable classification for real time and on-line application systems.

The proposed model in Section7.2.2 has not been implemented. Thus, implementing and evaluating it would be a potential interesting future work.

8 REFERENCES

[1] "www.twitch.tv."

[2] "www.xfire.com."

[3] R. H.264, "Advanced video coding for generic audiovisual services," *ITU-T Recommendation H.264*, May 2003.

[4] S. Saponara, C. Blanch, K. Denolf *et al.*, "The JVT advanced video coding standard: Complexity and performance analysis on a tool-by-tool basis," *IEEE Packet Video 2003, Nantes, France*, April, 2003.

[5] T. Stockhammer, M. M. Hannuksela, and T. Wiegand, "H.264/AVC in wireless environments," *IEEE Transaction, Circuits and System on Video Technology*, vol. 13 no.7, pp. 688 - 703, July, 2003.

[6] Z. Liu, L. Shen, and Z. Zhang, "An efficient intermode decision algorithm based on motion homogeneity for H.264/AVC," *IEEE transactions, circuits and systems for video technology*, vol. 19, pp. 128-132, 1 January, 2009.

[7] I. E. G. Richardson, *H.264 and MPEG-4 video compression*, April 2008 ed., England: John Wiley 2003.

[8] G. J. Sullivan, and T. Wiegand, "Video compression-from concepts to the H.264/AVC standard," *Proceedings of the IEEE*, vol. 93, pp. 18-31, 1 January, 2005.

[9] J.-W. Chen, C.-Y. Kao, and Y.-L. Lin, "Introduction to H.264 advanced video coding," in Asia and South Pacific Conference for Design Automation. , 2006, pp. 6.

[10] Y. Wu, Y. Zhao, and J. Li, "Brief analysis of the H.264 coding standard," in Third International Conference for Intelligent Information Hiding and Multimedia Signal Processing. IIHMSP. , 2007, pp. 154-157.

[11] T. Toivonen, "Efficient Methods for Video Coding and Processing, PhD," Department of Electrical and Information Engineering, University of Oulu, Linnanmaa, 2008.

[12] T. Wiegand, and G. J. Sullivan, "The H.264/AVC video coding standard," *IEEE Signal Processing Magazine*, pp. 148-153, March, 2007.

[13] J. Ostermann, J. Bormans, P. List *et al.*, "Video coding with H.264/AVC: tools, performance and complexity," *IEEE, Circuits and Systems Magazine,* vol. 4, no. 1, pp. 7-28, 2004.

[14] T. Wiegand, G. J. Sullivan, G. Bjontegaard *et al.*, "Overview of the H.264/AVC video coding standard," *IEEE Transactions, Circuits and Systems for Video Technology,* vol. 13, pp. 560-576, July, 2003.

[15] A. K. Khan, and H. Jamal, "The Intra prediction in H.264," *Novel Algorithms and Techniques In Telecommunications, Automation and Industrial Electronics,* pp. 11-15, 15 August, 2008.

[16] K. Seshadrinathan, R. soundararajan, A. C. Bovik *et al.*, "Study of subjective and objective quality assessment of video," *IEEE Transactions on Image Processing,* vol. 19, no. 6, pp. 1427 - 1441, 2010.

[17] "Methodology for the Subjective Assessment of the Quality of Television Pictures, Recommendation ITU-R BT. 500-11 ITU," 2002, pp. .

[18] S. Winkler, and R. Campos, "Video quality evaluation for Internet streaming applications," *SPIE Proceedings,* vol. 5007, pp. 104 - 115, 17 June, 2003.

[19] G. Zhai, J. Cai, W. Lin *et al.*, "Cross-dimensional perceptual quality assessment for low bit-rate videos," *IEEE Transactions on Multimedia,* vol. 10, no. 7, pp. 1316 - 1324, 7 November, 2008.

[20] "Objective perceptual video quality measurement techniques for digital cable television in the presence of a full reference, ITU-T Recommendation J.144," 2001.

[21] D. Marpe, T. Wiegand, and G. J. Sullivan, "The H.264/MPEG4 advanced video coding standard and its applications," *Communications Magazine, IEEE,* vol. 44, pp. 134-143, August, 2006.

[22] R. Vanam, E. A. Riskin, and R. E. Ladner, "H.264/MPEG-4 AVC encoder parameter selection algorithms for complexity distortion tradeoff," in IEEE on Data Compression Conference, 2009, pp. 372-381.

[23] M. Jiang, and N. Ling, "On Lagrange multiplier and quantizer adjustment for H.264 frame-layer video rate control," *IEEE Transactions, Circuits and Systems for Video Technology,,* vol. 16, pp. 663-669, May, 2006.

[24] Y.-K. Chen, A. Vetro, H. Sun *et al.*, "Optimizing INTRA/INTER coding mode decisions," in International Symposium on Multimedia Information Processing Conference, 1997, pp. 561-568.

[25] B. Schumitsch, H. Schwarz, and T. Wiegand, "Optimization of transform coefficient selection and motion vector estimation considering interpicture dependencies in hybrid video coding," *Proceedings- SPIE the international society for optical engineering,* vol. 5685, pp. 327-334, 2005.

[26] O. Nemcic, M. Vranjes, S. Rimac-Drlje *et al.*, "Comparison of H.264/AVC and MPEG-4 part 2 coded video," in ELMAR, 2007, Zadar, 2007, pp. 41 - 44.

[27] Recommendation ITU-R BT.601-5, "Studio encoding parameters of digital television for standard 4:3 and wide-screen 16:9 aspect ratios, ITU-T.," 1995.

[28] "http://livestream.com/," Accessed 1/11/2012.

[29] "http://www.justin.tv/," Accessed 15/09/2012.

[30] "http://www.gameranger.com/."

[31] "http://www.gamespyarcade.com/."

[32] "http://intl.garena.com/."

[33] "http://www.hlsw.org/."

[34] "http://www.kali.net/."

[35] "http://store.steampowered.com/."

[36] "http://www.linuxgames.com/xqf/index.shtml."

[37] Z. Hong, W. Cheng-ke, W. Yang-li *et al.*, "Fast mode decision for H.264/AVC based on macroblock correlation," in Advanced Information Networking and Applications, AINA 2005. 19th International Conference, 2005, pp. 775-780.

[38] J. Lee, and B. Jeon, "Fast mode decisions for H.264," in Multimedia and Expo, 2004. ICME '04. 2004 IEEE International Conference., 2004, pp. 1131-1134.

[39] G.-Y. Kim, and S.-H. Kim, "Fast mode decision algorithm for H.264 based on motion cost," in European Signal Processing Conference (EUSIPCO), 2005, pp. 1-4.

[40] H. Zeng, C. Cai, and K.-K. Ma, "Fast mode decision for H.264/AVC based on macroblock motion activity," *IEEE Transactions, Circuits and Systems for Video Technology* vol. 19, pp. 491-499, April, 2009.

[41] C.-H. Lee, C.-C. Lien, J.-L. Shih *et al.*, "A Fast macroblock mode decision algorithm for the baseline profile in the H.264 video coding standard," in Advances

in Image and Video Technology, Third Pacific Rim Symposium, PSIVT. Springer Berlin / Heidelberg, Japan, 2009, pp. 784-795.

[42] I. Choi, J. Lee, and B. Jeon, "Fast coding mode selection with rate-distortion optimization for MPEG-4 Part-10 AVC/H.264," *IEEE Transactions, Circuits and Systems for Video Technology,* vol. 16, pp. 1557-1561, December, 2006.

[43] J. Seung-Won, B. Seung-Jin, P. Chun-Su *et al.*, "Fast mode decision using all-zero block detection for fidelity and spatial scalable video coding," *IEEE Transactions, Circuits and Systems for Video Technology,* vol. 20, no. 2, pp. 201-206, 2010.

[44] W. Hanli, K. Sam, and K. Chi-Wah, "An Efficient mode decision algorithm for H.264/AVC encoding optimization," *IEEE Transactions on Multimedia,* vol. 9, no. 4, pp. 882-888, 2007.

[45] M. von dem Knesebeck, and P. Nasiopoulos, "An efficient early-termination mode decision algorithm for H.264," *IEEE Transactions on Consumer Electronics,* vol. 55, no. 3, pp. 1501-1510, 2009.

[46] Z. Tiesong, W. Hanli, S. Kwong *et al.*, "Fast mode decision based on mode adaptation," *IEEE Transactions, Circuits and Systems for Video Technology,* vol. 20, no. 5, pp. 697-705, 2010.

[47] Z. Huanqiang, M. Kai-Kuang, and C. Canhui, "Hierarchical intra mode decision for H.264/AVC," *IEEE Transactions, Circuits and Systems for Video Technology,* vol. 20, no. 6, pp. 907-912, 2010.

[48] S. Liquan, S. Yiwen, L. Zhi *et al.*, "Efficient SKIP mode detection for coarse grain quality scalable video coding," *Signal Processing Letters, IEEE,* vol. 17, no. 10, pp. 887-890, 2010.

[49] F. Pan, K. Choo, and L. M. Thinh, "Fast Rate-distortion optimization in H.264/AVC," in 9th international conference, KES, Melbourne, Australia, 2005.

[50] D. Zhang, Y. Shen, S. Lin *et al.*, "Fast inter frame encoding based on modes pre-decision in H.264," in Multimedia and Expo. ICME. IEEE International Conference on Publication, 2005, pp. 4.

[51] F. Pan, H. Yu, and Z. Lin, "Scalable fast rate-distortion optimization for H.264/AVC," *EURASIP Journal on Applied Signal Processing,* vol. 2006, pp. 1-10, 2006.

[52] X. He, P. Liu, K. Jia *et al.*, "A Fast intra-frame prediction algorithm based on the characteristic of macro-block and 2D-histogram for H.264/AVC standard," in Intelligent Information Hiding and Multimedia Signal Processing. Third International Conference on Publication 2007, pp. 182-185.

[53] T. D. H. Du, “Macroblock mode decision for H.264,” in International Multimedia Conference, Proceedings of the 7th ACM SIGMM international workshop on Multimedia information retrieval, Singapore, 2005, pp. 167-172.

[54] D. Kim, and J. Jeong, “A Fast mode selection algorithm in H.264 video coding,” in IEEE International Conference on Multimedia and Expo, 2006, pp. 1709-1712.

[55] C.-d. Shen, and S.-k. Li, “Fast prediction mode decision algorithm for H.264 based on hierarchical mode classification framework,” *Springer-Verlag Berlin Heidelberg, ISVC, LNCS 4292*, pp. 882 –890, 2006.

[56] Y. Liu, K. Tang, and H. Cui, “Efficient probability based macroblock mode selection in H.264/AVC,” in Visual Communications and Image Processing conference, SPIE, 2005, pp. 1.

[57] C. Chen-Kuo, P. Wei-Hau, H. Chiuan *et al.*, “Fast H.264 encoding based on statistical learning,” *IEEE Transactions, Circuits and Systems for Video Technology*, vol. 21, no. 9, pp. 1304-1315, 2011.

[58] S. Yu-Huan, and W. Jia-Ching, “Fast mode decision for H.264/AVC based on rate-distortion clustering,” *IEEE Transactions on Multimedia*, vol. 14, no. 3, pp. 693-702, 2012.

[59] L. Bumshik, and K. Munchurl, “An Efficient inter-prediction mode decision method for temporal scalability coding with hierarchical B-picture structure,” *IEEE Transactions on Broadcasting*, vol. 58, no. 2, pp. 285-290, 2012.

[60] Y.-K. Tu, J.-F. Yang, and M.-T. Sun, “Rate-distortion estimation for H.264/AVC coders,” in 2005 IEEE International Conference on Multimedia and Expo, 2005, pp. 4.

[61] Y.-K. Tu, J.-F. Yang, and M.-T. Sun, “Efficient rate distortion estimation for H.264/AVC coders,” *IEEE Transactions on circuits and systems for video technology*, vol. 16, pp. 600 - 611, May, 2006.

[62] H. Kim, and Y. Altunbasak, “Low-Complexity macroblock mode selection for H.264/Avc encoders,” in IEEE on Image Processing, 2004. ICIP apos;04. 2004 International Conference, 2004, pp. 765-768.

[63] M. Jeong Mee, and K. Jae Ho, “A New low-complexity integer distortion estimation method for H.264/AVC encoder,” *IEEE Transactions, Circuits and Systems for Video Technology*, vol. 20, no. 2, pp. 207-212, 2010.

[64] M. C. Chien, and P. C. Chang, “Optimal model-based complexity control for H.264 video encoding,” *Image Processing, IET*, vol. 6, no. 1, pp. 60-71, 2012.

[65] T.-Y. Kuo, and H.-J. Lu, “Efficient reference frame selector for H.264,” *IEEE transactions, circuits and systems for video technology,* vol. 18, pp. 400-405, 3 March, 2008.

[66] K. Lee, G. Jeon, and J. Jeong, “Fast reference frame selection algorithm for H.264/AVC,” *IEEE Transactions, Consumer Electronics,* vol. 55, no. 2, pp. 773-779, May, 2009.

[67] J. DongSan, and P. HyunWook, “An Efficient priority-based reference frame selection method for fast motion estimation in H.264/AVC,” *IEEE Transactions, Circuits and Systems for Video Technology,* vol. 20, no. 8, pp. 1156-1161, 2010.

[68] R. Jillani, and H. Kalva, “Low complexity intra MB encoding in AVC/H.264,” *IEEE Transactions on Consumer Electronics,* vol. 55, pp. 277-285, February, 2009.

[69] S.-H. Ri, Y. Vatis, and J. Ostermann, “Fast inter-mode decision in an H.264/AVC encoder using mode and Lagrangian cost correlation,” *IEEE transactions, circuits and systems for video technology,* vol. 19, pp. 302-306, February, 2009.

[70] M. Paul, M. R. Frater, and J. F. Arnold, “An Efficient mode selection prior to the actual encoding for H.264/AVC encoder,” *IEEE Transactions on Multimedia,* vol. 11, no. 4, pp. 581-588, 2009.

[71] T. Qiang, and P. Nasiopoulos, “Efficient motion re-estimation with rate-distortion optimization for MPEG-2 to H.264/AVC transcoding,” *IEEE Transactions, Circuits and Systems for Video Technology,* vol. 20, no. 2, pp. 262-274, 2010.

[72] T. Yih Han, L. Wei Siong, T. Jo Yew *et al.*, “Complexity scalable H.264/AVC encoding,” *IEEE Transactions, Circuits and Systems for Video Technology,* vol. 20, no. 9, pp. 1271-1275, 2010.

[73] Z. Zhuo, and L. Ping, “A Statistical analysis of H.264/AVC FME mode reduction,” *IEEE Transactions, Circuits and Systems for Video Technology,* vol. 21, no. 1, pp. 53-61, 2011.

[74] J. L. Nunez-Yanez, A. Nabina, E. Hung *et al.*, “Cogeneration of fast motion estimation processors and algorithms for advanced video coding,” *IEEE Transactions, Very Large Scale Integration (VLSI) Systems,* vol. 20, no. 3, pp. 437-448, 2012.

[75] K. Soon-kak, A. Punchihewa, D. G. Bailey *et al.*, “Adaptive simplification of prediction modes for H.264 intra-picture coding,” *IEEE Transactions on Broadcasting,* vol. 58, no. 1, pp. 125-129, 2012.

[76] X. Lu, and G. R. Martin, “Fast H.264/SVC inter-frame and inter-layer mode decisions based on motion activity,” *Electronics Letters,* vol. 48, no. 2, pp. 84-86, 2012.

[77] M. Claypool, K. Claypool, and F. Damaa, “The effects of frame rate and resolution on users playing first person shooter games,” in Proceedings of ACM/SPIE Multimedia Computing and Networking (MMCN), San Jose, California, USA, 2006.

[78] K. Claypool, and M. Claypool, “The Effects of resolution on users playing first person shooter games,” in Proceedings of ACM/SPIE Multimedia Computing and Networking (MMCN), San Jose, California, USA, 2007.

[79] M. Claypool, and K. Claypool, “Perspectives, frame rates and resolutions: It's all in the game,” in Proceedings of the 4th ACM International Conference on the Foundations of Digital Games (FDG), Florida, USA, 2009.

[80] G. J. F. Smets, and K. J. Overbeeke, “Trade-off between resolution and interactivity in spatial task performance,” *Computer Graphics and Applications, IEEE,* vol. 15, no. 5, pp. 46 - 51, September 1995.

[81] C. Deng, W. Lin, B.-s. Lee *et al.*, “Comparison between H.264/AVC and Motion jpeg2000 for super-high definition video coding,” in Image Processing (ICIP), 2010 17th IEEE International Conference on, 2010, pp. 2037 - 2040.

[82] T. Wiegand, H. Schwarz, A. Joch *et al.*, “Rate-constrained coder control and comparison of video coding standards,” *IEEE Transactions, Circuits and Systems for Video Technology,* vol. 13, no. 7, pp. 688 - 703, 2003.

[83] M. Brown, D. Bushmitch, K. Kerpez *et al.*, “Low-bit rate video codec parameter evaluation and optimization,” in Military Communications Conference, MILCOM 2009. IEEE, 2009, pp. 1 - 20.

[84] J. D. McCarthy, M. A. Sasse, and D. Miras, "Sharp or smooth?: comparing the effects of quantization vs. frame rate for streamed video," *Conference on Human Factors in Computing Systems,* [April 24 - 29, 2004].

[85] A. Bhat, S. Kannangara, Y. Zhao *et al.*, “A Full reference quality metric for compressed video based on mean squared error and video content,” *IEEE Transaction, on Circuits and Systems for Video Technology,* vol. 22, no. 2, February, 2012.

[86] S. Winkler, and P. Mohandas, “The Evolution of video quality measurement: from PSNR to hybrid metrics,” *IEEE Transactions on Broadcasting.,* vol. 54, no. 3, pp. 660-668, September, 2008.

[87] Q. Huynh-Thu, M.-N. Garcia, F. Speranza *et al.*, “Study of rating scales for subjective quality assessment of high-definition video,” *IEEE Transactions on Broadcasting*, vol. 57, no. 1, pp. 1 - 14, March, 2011.

[88] "http://www.fraps.com/."

[89] T. Stutz, and A. Uhl, “A Survey of H.264 AVC/SVC encryption,” *IEEE Transactions, Circuits and Systems for Video Technology*, vol. 22, no. 3, pp. 325-339, 2012.

[90] http://www.netindex.com/, Accessed 2/12/2011.

[91] http://media.ofcom.org.uk/, Accessed 2/12/2011.

[92] http://www.ispreview.co.uk/, Accessed 2/12/2011.

[93] http://www.ffmpeg.org/, Accessed 5/01/2012.

[94] ITU-T, “Advanced video coding for generic audiovisual services,” *ITU-T Recommendation H.264*, May, 2003.

[95] G. W. Snedecor, and W. G. Cochran, *Statistical Methods*, , 8th ed.: Ames, IA: Iowa State University, 1989.

[96] G. J. Sullivan, J. R. Ohm, W. J. Han *et al.*, “Overview of the high efficiency video coding (HEVC) standard,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1649-1668, 2012.

[97] I. K. Kim, J. Min, T. Lee *et al.*, “Block partitioning structure in the HEVC standard,” *IEEE Transactions, Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1697-1706, 2012.

[98] D. Brezeale, and D. J. Cook, “Automatic video classification: a survey of the literature,” *IEEE Transactions, Systems, Man and Cybernetics, Part C: Applications and Reviews.*, vol. 38, no. 3, pp. 416-430, 2008.

[99] R. W. Lienhart, “Comparison of automatic shot boundary detection algorithms,” *In Procedings, SPIE Conference Storage Retrieval Image Video Databases VI*, vol. 3656, pp. 209 - 301, 1999.

[100] R. S. Jadon, S. Chaudhury, and K. K. Biswas, “A fuzzy theoretic approach for video segmentation using syntactic features,” *Pattern Recognition Letter*, vol. 22, no. 13, pp. 1359 - 1369, 2001.

[101] T. Ba Tu, and C. Dorai, "Automatic genre identification for content-based video categorization," in Proceedings, 15th International Conference on Pattern Recognition. , 2000, pp. 230-233 vol.4.

[102] W. Peng, C. Rui, and Y. Shi-Qiang, "A hybrid approach to news video classification multimodal features," in Proceedings of the Joint Conference of the Fourth International Conference on Information, Communications and Signal Processing, Fourth Pacific Rim Conference on Multimedia, 2003, pp. 787-791 vol.2.

[103] Y. Xun, L. Wei, M. Tao *et al.*, "Automatic video genre categorization using hierarchical SVM," in Image Processing, 2006 IEEE International Conference on, 2006, pp. 2905-2908.

[104] N. Dimitrova, L. Agnihotri, and G. Wei, "Video classification based on HMM using text and faces," in Europe Signal Processing Conference (EUSIPCO 2000), Tampere, Finland, 2000.

[105] J. L. Barron, D. J. Fleet, and S. S. Beauchemin, "Performance of optical flow techniques," *International Journal of Computer Vision,* vol. 12, no. 1, pp. 43 - 77, 1994.

[106] H. B. K. P., and S. B. G., "Determining optical flow," *Artificial Intelligence,* vol. 17, no. 1/3, pp. 185-203, 1981.

[107] M. Roach, J. Mason, and X. Li-qun, "Video genre verification using both acoustic and visual modes," in Proceeding IEEE Workshop, Multimedia Signal Processing, 2002, pp. 157-160.